DISCOVERING GOVERNING EQUATIONS FROM NOISY AND INCOMPLETE DATA

A Dissertation
Presented to
The Academic Faculty

By

Patrick A.K. Reinbold

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Physics

Georgia Institute of Technology

December 2020

DISCOVERING GOVERNING EQUATIONS FROM NOISY AND INCOMPLETE DATA

Approved by:

Dr. Roman O. Grigoriev, Advisor
School of Physics
Georgia Institute of Technology

Dr. Michael F. Schatz
School of Physics
Georgia Institute of Technology

Dr. Kurt Wiesenfeld
School of Physics
Georgia Institute of Technology

Dr. Molei Tao
School of Mathematics
Georgia Institute of Technology

Dr. Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Date Approved: January 11, 2000

ACKNOWLEDGEMENTS

First and foremost, I'd like to thank my parents; without them, I wouldn't exist. Moreover, the wisdom, skills, and values they have imparted in me throughout the years has in large part made me who I am. Secondly, I acknowledge the plethora of instructors that have strongly influenced me throughout my life. Being a student for two decades, one accumulates a number of mentors. Two educators who helped me mature as a person are Chris Detweiler and Adam Kicklighter; without their guidance, I would have a very different outlook on life. Thank you to my undergraduate advisors, Faramarz Farahi, Thomas Suleski, Nathan Fried, and Donald Jacobs. My years in your classes, labs, and under your advisement fostered my skills as a researcher and thinker.

I must also acknowledge my peers that enriched my life while at Tech. There are my predecessors, Balachandra Suri and Ravi Pallantla, who guided me when I was new to the group. They have also expanded my knowledge as I continued my research path, sacrificing their time for me even after graduating. There are also my peers that helped me adjust to a new city during my first year, and have provided me with good times during my stay: James McInerny, Alisdair Gent, Boli Zhou, Chao Li, Michael Ryan, and Lin Xin. Talking with Daniel Gurevich was always a pleasure, and Matt Gudorf proved to be a great office mate and friend. Of course, my roommate and primary colleague, Logan Kageorge, always made my domestic and professional life more enjoyable. Last but not least, thank you to the two professors that were most influential during my graduate career: Mike Schatz and Roman Grigoriev. Your criticism, encouragement, and insight were invaluable to my academic development. I was provided with a number of opportunities that were great experiences, and the skills and knowledge I have gained during my time at Tech are largely thanks to your tutelage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Partial differential equations (PDEs) provide macroscopic descriptions of systems in numerous fields, such as physics, biology, and chemistry. Additionally, with increasingly vast amounts of data becoming available with the advancement of technology, machine learning is now offering an alternative to traditional model construction (eg. from first principles). This alternative is particularly attractive for systems that are too complex for derivation from first principles to be tractable, or worse, where no first principles are known at all. The research presented in this thesis advances the current state of data-driven PDE modeling. The fundamental approach involves converting a candidate PDE into a system of algebraic equations linear in model parameters via a carefully designed evaluation procedure, and then using sparse regression to narrow down to the model that best fits the data. Existing regression methods, when applied to PDEs, rely on linear systems that are sensitive to noise corruption in the observations. However, regression methods can quickly explore the fitness of many different model options, and so are the default choice for systems that could be described by a large set of potential models.

Several data-driven approaches to model discovery have been designed recently, but most have been shown to be inadequate for application to high-dimensional data described by PDEs. Some approaches (like training a neural network to 'learn' the dynamics, from which the model can be backed out of the network parameters) are less susceptible to noise corruption, but are currently ill-suited to finding the best model out of a large set of candidates, since it does not directly accomodate changes to the model structure. On the other hand, symbolic regression methods of the type mentioned above can discern between different potential models efficiently. They do well for low-dimensional systems described by ordinary differential equations (ODEs), but have yet to see successful application to PDEs for experimental data. The main problem is that constructing the linear system prior to performing the regression has traditionally required locally evaluating derivatives of discrete

and noisy data. Derivatives (especially of higher order) computed from discrete and noisy data is notoriously inaccurate and has hindered many attempts at PDE model identification.

Besides relying on local (inaccurate) derivative information, existing regression methods rarely constrain the library of candidate model terms. However, it is found here that utilizing knowledge of relevant physics and symmetry can ensure that the candidate terms are both interpretable and not un-physical. The main breakthrough of this thesis is developing an alternative approach to building the linear system fed to the regression algorithm. By considering the weak form of the candidate model, the derivatives on many model terms can be moved onto a weight function whose derivatives are analytically known. The linear system is then filled with integral values that are far less sensitive to noise corruption. Furthermore, the weight function can be carefully designed to remove dependence on certain latent variables from the weak form, which enables identification of PDEs that would otherwise be impossible to work with due to the missing information. These latent variables can later be reconstructed using the available data, domain knowledge, and (crucially) the model identified using the weak formulation. The preceding approach was tested on a number of synthetic examples, and then applied to experimental turbulent fluid flow data to obtain a 2D model consistent over a range of driving values. In summary, the research presented here develops and validates a methodology for data-driven discovery of PDEs that is robust to noise and latent variables, and it demonstrates the ability to do so on real-world data for the first time.

CHAPTER 1
INTRODUCTION AND BACKGROUND

One of the main pursuits of theoretical science is the development of proper models that enable prediction and provide physical insight. Traditionally, model construction almost always requires understanding of the physics of what is being described. The constructed models will undoubtedly contain some number of parameters, and depending on the scenario, determining them can be difficult. If the physical insight is incomplete, or worse, wrong, the crafted model is likely not a good one. The established way to overcome incomplete physical insight is to interact between theory and experiment in an iterative fashion to improve the generality and accuracy of models.

For example, consider Carl Friedrich Gauss's determination of the orbit of the planetoid Ceres. Other formidable thinkers, such as Laplace, were incapable of providing an accurate prediction of Ceres's orbit based on their limited understanding of the physics. Gauss was able to engineer an approach that compensated for deficiencies in physical understanding by utilizing the known physics of related problems (such as the motion of the moon), heuristic considerations, and by leveraging the observations themselves with mathematical tools (his method of least squares) [1, 2].

The aim of this work is in the same spirit of letting the physical measurements decide the model for themselves, with gentle guidance from existing physical understanding of the problem. This is in contrast to the established paradigm, which might require, due to its iterative nature, many experiments and theoretical insights. This thesis also stands in contrast to a purely data-driven method, which without the guidance of physical insight could lose its interpretability or spend too much time exploring unphysical models. The idea behind the hybrid approach presented here is that it will not only provide a model that enables quantitative prediction of the system without complete knowledge of the physics

beforehand, but also aid in the discovery of the underlying physics.

The following section will briefly explain the hybrid learning procedure presented in this thesis. Additionally Section 1.1 will also describe the critical issues facing the state of the art, particularly with regard to discovering PDE models. Afterward, Section 1.2 will present alternatives to the linear regression paradigm employed by the hybrid learning procedure. Then, Section 1.3 will motivate overcoming the difficulties in PDE model discovery by showing a how ubiquitous PDE models are.

1.1 Problem Overview

Before jumping into any particular examples in the following chapters, a brief conceptualization of the general system identification problem is given. Here notation and terminology will be set for the remainder of the thesis. First, consider a general representation of a first-order differential equation

$$\partial_t \mathbf{u} = \sum_{n=1}^{N} c_n \mathbf{F}_n[\mathbf{u}, \nabla, \dots], \tag{1.1}$$

where c_n are constant coefficients, $\mathbf{F}_n$ represent nonlinear combinations of derivatives and state variables, and N is the number of terms in the model. All the equations considered here will be of this type, but the procedure can be adapted to other equations by separating a different term to the left-hand side rather than the first time-derivative. The governing equation (1.1) is the object that symbolic regression seeks to identify.

To do so, first, the "library" $\{\mathbf{F}_n\}_n$ of candidate terms must be constructed. This library acts as the basis of a space that an accurate model exists in. If a model does not formally exist within the library-space, then it might be at least well represented; for example consider

the small angle approximation for a physical pendulum:

$$\ddot{\theta} = c_1 \sin(\theta) \tag{1.2}$$

$$\approx c_1 \theta. \tag{1.3}$$

A library with only finite-order polynomials in θ doesn't fully capture the dynamics, but it would yield a model that is fairly accurate under the condition that θ is small. Using a hybrid approach, the library should be constructed to employ available knowledge of the system such as relevant physical laws, phenomenology, or necessary symmetries. Using domain knowledge to constrain and construct the library is treated in more depth in Chapter 3.

Once a suitable library of N terms has been constructed, the next step is to formulate symbolic regression of (1.1) as a solution to the linear system:

$$\partial_t \mathbf{u} = \sum_n c_n \mathbf{F}_n$$

$$\Downarrow$$

$$\mathbf{q}_0 = Q\mathbf{c}, \tag{1.4}$$

where $Q = [\mathbf{q}_1, \mathbf{q}_1, \dots \mathbf{q}_N]$ is a $K \times N$ matrix and shall be referred to as the "library matrix". The columns of the library matrix correspond to the different potential terms of the model, while the rows correspond to unique equations linear in the coefficients c_n. One way to get an ensemble of unique equations linear in c_n is to evaluate the model terms $\mathbf{F}_n$ at different locations in space and time. The terms need to be computed at $K \geq N$ different locations in order for the coefficients to be determined.

To make this more concrete, consider the Kuramoto-Sivashinsky equation

$$\partial_t u = c_1 u \partial_x u + c_2 \partial_x^2 u + c_3 \partial_x^4 u. \tag{1.5}$$

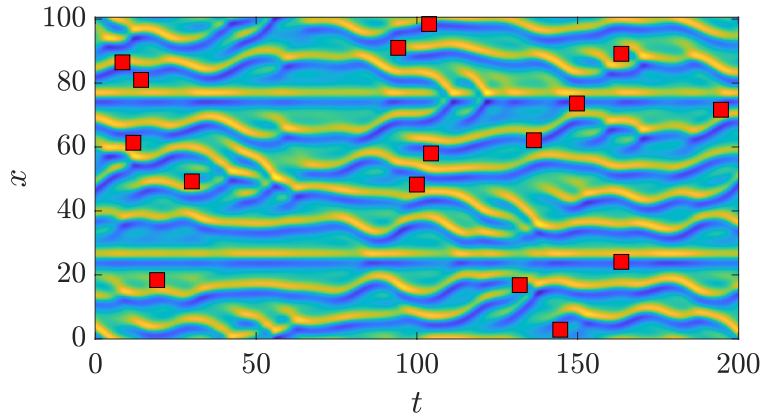


Figure 1.1: A solution to (1.5) with $c_1 = c_2 = c_3 = -1$. A random sampling to represent points in space time used to create a library matrix is represented by red squares.

Rather than knowing c_1, c_2, c_3 and solving for u, in this case we are instead interested in the inverse. To convert this from a differential equation in u into an algebraic equation linear in c_n, the model terms need be evaluated at some point in space in time. For example, at a particular point the equation could reduce to

$$-0.061 = 0.863c_1 - 1.378c_2 + 0.578c_3. \tag{1.6}$$

The sampling would be repeated $K \geq 3$ times, as shown in Fig. 1.1, in order to create the linear system (1.4). If the form of (1.5) isn't known ahead of time, then additional terms would have to be evaluated; in this case the library could include polynomial combinations of u and derivatives in x.

After a library matrix has been constructed, finding a model reduces to finding a $\mathbf{c}$ that satisfies (1.4). The simplest way to do this is via least squares. This has been done by many to identify a dynamical model [3, 4, 5, 6, 7]. However, the general least-squares solution assigns non-zero values to c_n that should technically not be present in the model, even if the c_n assigned is very small. To avoid this overfitting, all possible subsets of Q could be compared, and the model that balances parsimony with accuracy could be selected by hand or by some metric (e.g., Akaike Information Criterion). However, when the number of columns in Q become large, this is not feasible. One alternative is to employ a back-

ward elimination scheme that iteratively removes columns from Q that increase the residual $\|Qc - q_0\|$ the least when they are removed and stops once no terms increase the residual below some threshold [8]. Another related approach is to use the magnitude of c_n (scaled by $\|\mathbf{q}_n\|$) to quantify the relevance of $\mathbf{F}_n$ to the observed data, and to remove $\mathbf{q}_n$ from the library if the contribution is below some threshold. Then, the reduced $\mathbf{c}$ is computed, and the process is repeated until no terms are deemed insignificant. This is popularly known as SINDy [9], but a very similar algorithm was developed earlier as "structure selection" [10].

These sequential thresholding approaches are heuristic alternatives to regression techniques that regularize the least squares minimization with different, but similar penalty terms. They can be written as

$$\min_{\mathbf{c}} \|\mathbf{q}_0 - Q\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_s, \tag{1.7}$$

where Q_k denotes the k^{th} row of the library matrix and s is some integer. Choosing $\|\cdot\|_s = \|\cdot\|_0$ enforces model sparsity by penalizing non-zero elements of $\mathbf{c}$ [11]. A similar effect can be achieved with the well-known LASSO, which corresponds to $\|\cdot\|_s = \|\cdot\|_1$ [12, 13, 14]; solutions for $\mathbf{c}$ are encouraged to be identically 0 using the LASSO, and so can yield sparse (easily interpretable) models. Finally, $\|\cdot\|_s = \|\cdot\|_2^2$ is known as ridge regression [15]; although regularizing with an L^2 penalty can help prevent overfitting when columns of Q are collinear, it does not generally enforce sparsity in $\mathbf{c}$. The regularization methods (1.7) are generally solved using gradient descent and are thus more computationally expensive than the heuristic algorithms. Interestingly, the heuristic algorithms can be viewed as approximations to the regularization approaches, such as the sequential thresholding algorithm SINDy approximating L^0 regularization [16].

It is also worth mentioning "entropic regression" [17], which fares favorably to both heuristic and regularized regression. This approach employs iterative ordinary least squares, but it chooses its subspace of columns of Q based on maximizing the mutual information

between the columns $\mathbf{q}_n$ and $\mathbf{q}_0$. Many of these regression methods yield satisfactory results for reasonably large K and reasonably small errors in the elements of the library matrix q_{kn}. However, the latter condition is rarely satisfied for local evaluations of the model terms like in (1.6).

This inability to evaluate q_{kn} with acceptable accuracy under "realistic conditions," sparse grids and noise corruption, has been observed historically in experimental [4, 7] and synthetic [3, 4, 8, 18, 10, 19, 20] data. The cause of the difficulty becomes clear when we consider $\mathbf{F}_n$ that involve derivatives, such as $\partial_x u$. A single point in space is insufficient to evaluate such a library term, but rather local neighborhoods are required. To approximate $\partial_x u$, *at least* two points separated in x are needed. For example, the forward finite difference scheme

$$\partial_x u = \frac{u(x + \Delta x) - u(x)}{\Delta x} + \varepsilon_d \Delta x, \tag{1.8}$$

where ε_d is the error due to discretization and Δx is the distance between adjacent observations, requires knowing u at x and $x + \Delta x$. For both simulations and experiment, discretization is unavoidable, and this will always incur errors in the estimations of the derivatives. The error due to discretization can be decreased by increasing the measurement density, but doing so presents challenges for both experiment and simulation.

Furthermore, experiments contain unavoidable measurement error. Consider as an illustrative example additive error

$$\tilde{u} = u + \chi, \tag{1.9}$$

where $\tilde{u}$ is the measured variable, u is the "truth", and χ is the noise corruption, then using the finite difference scheme (1.8) to estimate the first-derivative gives

$$\frac{\tilde{u}(x + \Delta x) - \tilde{u}(x)}{\Delta x} = \partial_x u + \frac{\chi(x + \Delta x) - \chi(x)}{\Delta x} + \varepsilon_d \Delta x \tag{1.10}$$

$$= \partial_x u + \varepsilon_n \frac{1}{\Delta x} + \varepsilon_d \Delta x, \tag{1.11}$$

where ε_d is the error due to discretization and ε_n is the error due to the measurement noise. Since $\Delta x < 1$, taking derivatives amplifies the measurement errors ε_n. The amplification due to noise can be diminished by increasing Δx, but this comes at the cost of amplifying the discretization errors ε_d. Choosing higher-order finite-difference schemes can improve the cumulative error [8]; unfortunately, even the optimal choice of Δx that balances discretization and noise error yields accuracy insufficient for model reconstruction. To make matters worse, higher derivatives amplify the measurement noise even more. This is why ODEs have seen success with this approach (they usually involve a single first-order time-derivative), but PDEs have seen almost no successful experimental application.

Although useful for illustrating the fundamental problem, finite differences (of any order) is a poor choice for noisy data, and better techniques to estimate the derivatives exist. For instance, a method to smooth, interpolate, and take derivatives of the data in one-dimension was developed by Savitzky and Golay [21]. They fit polynomials to the data, and computed the derivative(s) using the polynomial coefficients. The method can exploit many observations in the neighborhood of the point of interest and is more robust to noise corruption than finite differences. The "Savitzky-Golay filter" can be extrapolated for use in higher dimensions, but its accuracy is ultimately limited as well. Further limitations of local polynomial estimates of derivatives in the context of system identification is shown in Chapter 3. Using a global method such as a Fourier operator to take derivatives exhibits similar sensitivity to noise [10], due to the Fourier operator disproportionately amplifying high frequency noise. Regularization methods (e.g., Tikhonov [22, 23, 24] and Total Variation [25, 24]) yield the most reasonable derivative estimates in the presence of noise (due to enforcing smoothness on the solution), though they are generally more expensive numerically and still have limited accuracy for higher-order derivatives. Other derivative estimation techniques for noisy data are given in [24], but it is clear that for higher-order equations, there is just no way to construct the library elements with acceptable accuracy.

Instead of trying to find better ways to estimate derivatives from noisy data, they can be

avoided entirely. To do this, use the weak form of (1.1), rather than the strong form. For example, consider again the Kuramoto-Sivashinsky equation, this time in its weak form

$$\langle w, \partial_t u \rangle_\Omega = c_1 \langle w, u \partial_x u \rangle_\Omega + c_2 \langle w, \partial_x^2 u \rangle_\Omega + c_3 \langle w, \partial_x^4 u \rangle_\Omega, \tag{1.12}$$

where $\langle w, \cdot \rangle_\Omega = \iint_\Omega w \cdot dxdt$ and $w(x,t)$ is called a weight function. Evaluating the integrals over some definite integration domain Ω yields an equation that is still linear in the coefficients c_n; the rows of the library matrix can then be filled by considering an ensemble of unique combinations of Ω and w. Integration by parts (with a weight function that kills boundary terms) allows derivatives to be transferred from the noisy data to a weight function whose derivatives are known exactly. Using the weak form to construct Q to avoid derivative noise amplification was originally used by Shinbrot in 1954 for ordinary differential equations [3], adapted to partial differential equations by Perdreauville and Goodson in 1966 [4], and it has been further adapted, re-purposed, and re-discovered many times since [26, 27, 28, 29, 30, 31, 32, 33, 34]. Using a weak formulation for library construction will be discussed in more detail in Chapters 4 and 5.

A final problem that linear regression methods face is that not all variables relevant to construct Q can be measured. If these "latent variables" have certain exploitable properties (e.g., time-independence) then they can be "projected out" of the governing equations (e.g., by applying a time-derivative). Two different examples are presented in Chapters 3 and 5, but in general some "projection" has to be devised to remove latent variables for a linear regression method to work for model discovery.

1.2 Alternatives to Linear Regression Methods

There also exists several methods for discovering the governing equations that don't involve the constrained linear regression methods discussed above. The first alternative to mention is that of using genetic or evolutionary optimization algorithms to search the space of

mathematical expression to find a model that best fits the data. For instance, in [35, 36], an ensemble of models was initialized, each with a random form. Each model in the ensemble was simulated to compare with observations. Then, each underwent a random perturbation to the form of their model. The perturbations comprise applying new operators or including more variables into the model term(s). If the perturbation improved the agreement with the observation, then those "genes" were advanced; if not, then the prior model was kept. The benefit of this approach is that it assumes essentially nothing about the structure of the model. However, its validation procedure involves simulating trajectories at each iteration, which is tractable for low-dimensional models (ODEs), but not a good option for higher-dimensional models (PDEs). Furthermore, the forms of many PDEs are restricted by the symmetries and physical mechanisms of the system they describe, and so the mathematical expressions allowed in the governing equation cannot be arbitrary. As such, stochastic optimization algorithms are likely to waste a lot of time considering models that intuition could have culled ahead of time.

Nonlinear regression is also an alternative. One such technique, alternating conditional expectations (ACE) [37], has been applied to "learn" ODEs [38] and PDEs [39, 40]. The method consists of labeling certain combinations of fields and their derivatives as independent variables, (e.g. for the Kuramoto-Sivashinsky equation $v_0 = \partial_t u$, $v_1 = u\partial_x u$, $v_2 = \partial_x^2 u$, and $v_3 = \partial_x^4 u$), and then finding the functions $\theta(v_0), \phi_i(v_i)$ that minimize the error

$$\mathbb{E}\big[(\theta(v_0) - \sum_i \phi_i(v_i))^2\big]. \tag{1.13}$$

The ACE algorithm is capable of producing arbitrary nonlinear functions $\phi_i(v_i)$, assuming that there are enough observations and the v_i can be estimated with sufficient accuracy. There are two main problems with this approach, when applied to the identification of partial differential equations. First, evaluating v_i requires the estimation of derivatives, meaning that higher-order equations will be sensitive to noise corruption. This was found to be the case when a complex Ginzburg-Landau equation (second-order in space) was discov-

ered from experimental binary fluid convection data [40]. Second, though ACE can find nonlinear transformations ϕ_i, how to choose the $v_i(u, \partial_x, \dots)$ such that they correspond to distinct physical mechanisms isn't clear unless information about the system is already known.

Another alternative, Data assimilation [41] is traditionally used as a method to estimate the state of systems where there is considerable uncertainty in the truth of the state. Meteorology and oceanography are popular examples, where the systems are vast, difficult to measure densely, and have frequent external perturbations. Data assimilation seeks to find the optimal linear combination of observations and predictions from a known model. Two common ways to optimize the linear combination are variational approaches and ensemble filters [42]. By optimally combining real-world observations and the results of simulations, predictions are made more accurate. The state-estimation procedure can be modified such that the model parameters are also identified. This is accomplished by appending the desired model parameters to the "state" vector and then solving a more complicated optimization problem. This procedure has been applied to synthetic data generated from a PDE [43]; for a variety of grid densities, noise corruption magnitudes, and initial guesses, the method yielded predictions close to the "truth" (though time-averaging was needed to make the estimates converge in the noise-corruption case). One downside of this approach is its computational cost – the system must be simulated and a variational method must be used for each time step, until the parameter estimates converge. Another downside is that although the parameters are being identified, the form of the model is assumed known.

Instead of providing estimates for both the state and the parameters, variational methods can be used to identify model parameters directly. This is done by minimizing the difference between model predictions and observations. This has been applied to PDEs with constant coefficients [44, 38, 45, 18] and for spatially dependent model parameters [46]. One of the major boons is that the method is quite robust to noise corruption, measurement fidelity, and type of measurement [18]. For instance, the measurements need not

be densely measured, be directly measured (e.g. measurements could be local averages), or even contain all variables. A trajectory is simulated using a best-guess model, the state is transformed as needed to match the kind of measurements available, and new parameters are chosen using a variational procedure. A problem this approach can run into, is that for chaotic dynamics convergence can be very slow if the trajectory is very long, since tiny differences in initial conditions and parameter guesses can yield significantly different dynamics after enough time. A solution to this is multi-shooting [45]. Another fundamental drawback is that, like data assimilation, the model form must already be known, along with the boundary conditions. Furthermore, the method is much more computationally expensive than regression counterparts [18]. Finally, unlike the regression approaches which require no simulations, the variational approaches require boundary conditions to be known.

Yet another alternative is Gaussian Process Regression (GPR)[47, 48]. Although it is labeled as "regression," it is sufficiently different that it is considered as an alternative. The method begins by making an assumption that the dependent variable at each location in space is a Gaussian process (GP) spatially correlated to its neighbors. Next, it builds upon the analytical property of Gaussian processes that the output of a linear system whose input is Gaussian distributed is again Gaussian. This means that adjacent states in time are together a Gaussian process. The covariance matrix of this multi-output GP then encodes the physical laws that connect the two states in time (the partial differential governing equations). Then, if one has access to two states near in time, the hyperparameters of the covariance matrix (which are the coefficients of the governing equation) can be learned by minimizing the negative log marginal likelihood. A benefit of this approach is that the derivatives contained in the governing equations are applied to the covariance functions (such as a squared exponential), rather than the data themselves. It is likely that for this reason that the parameter estimates are robust to noise. Another boon of GPR is that the sampling locations are arbitrary. However, there are three drawbacks to this approach. First,

to solve for the parameters requires the inversion of dense, large covariance matrices which scales cubicly with the number of observations used [47]. More crucially, this approach assumes a known form of the PDE ahead of time. Finally, the property that the output of a system is Gaussian if the input is Gaussian is only true for *linear* systems; hence, the governing equations need to be linearized before being used to construct the variance matrix.

A final alternative worth mentioning is using neural networks to learn the governing equations. The key idea is approximate the solution to the governing PDE by a neural network (or many). The neural net is trained by minimizing the differences between its approximation of the solution and observations as well as, crucially, a constraint term that enforces the governing equations [49]. Other properties can be enforced via constraint terms as well, such as boundary conditions. Not only are the neural networks trained to approximate the staet, but the parameters of the physical constraint term, which correspond to the coefficients in the governing equation, are also learned. The learned parameters exhibit good resilience to noise corruption of the observations, or at least better than regression methods that rely on direct evaluation of the derivatives. Furthermore, they can learn the governing equations even in the absence of latent variables, and they are indifferent to observation locations. A downside, like many of the alternatives here is that the model form is assumed to be known ahead of time. The method can be extended so that it uses a large number of model terms with variable coefficients, with the hopes that the terms that don't describe the data will have their coefficients set very small at every point in space [50]. Unfortunately, this lacks the physical interpretability that methods like LASSO and SINDy provide due to them setting small terms conclusively to 0. As such, despite being compeling parameter estimators and methods of prediction, their lack of model selection mechanism makes them less suited to symbolic regression.

1.3 Systems Described by PDEs

Above, several approaches to model discovery are detailed. A commonality between many of them is that they were derived initially for application to low-dimensional systems (ODEs). Unfortunately, many of the techniques that work for low-dimensional systems do not generalize well to high-dimensional systems (PDEs). One such example is if the system (or an adjoint with comparable dimensionality) needs to be simulated repeatedly to refine the model estimate; while this may be fine for systems of ODEs, this becomes prohibitive for use with PDEs. One of the main appeals of the linear regression methods focused on in Section 1.1 is that it scales well with increased system size.

However, a unique problem arises for PDEs: derivatives. For many ODEs, there is only one term with a derivative, and it is often only order 1 or 2. For PDEs, every term could contain a derivative, and the derivatives can be fourth-order, sometimes even higher. As shown above, this spells disaster if the data are corrupted with even minuscule amounts of noise; unfortunately, noise corruption of some form is unavoidable for experimental data. A weak formulation of the problem was presented as an alternative, and much of the thesis will elaborate further. This section will be focused on emphasizing why the weak formulation is worth developing, as opposed to just accepting not being able to apply symbolic regression to PDEs.

Many important quantities of interest are dependent on extended variables (space and time); i.e. they can be represented as (potentially multi-component) fields. Moreover, our universe is not boring and these fields exhibit spatiotemporal dynamics (ie, they are not uniform in space and time), due (mostly) to systems existing outside equilibrium [51]. The structure of these fields in time and space is often governed by a partial differential equation (PDE). There are plenty of fundamental PDEs which need no introduction, like Maxwell's equations and Schrödinger's equation. Another example, the Navier-Stokes

equations (NSE) in non-dimensional form

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f}, \tag{1.14}$$

are used to describe conservation of momentum of a fluid, where $\mathbf{u}$ is the fluid velocity, p is the fluid pressure, Re is a non-dimensional parameter that can be thought of as a measure of the balance between inertia and dissipation called the Reynolds number, and $\mathbf{f}$ represents the net body forces acting on the fluid. However, PDE modeling isn't limited to fundamental *physical* laws; they are found in a variety of disciplines. A few low-dimensional (in both independent and dependent variables) examples are given below to illustrate how broad the applications are.

One of the simplest examples involving a quadratic nonlinearity is Fisher's (or sometimes Kolmogorov-Petrovsky-Piskunov, or Fisher-KPP) equation

$$\partial_t u = \partial_x^2 u + au(1-u) \tag{1.15}$$

which was first developed independently by Fisher [52] and by Kolmogorov, Petrovsky, and Piskunov [53] to describe the spatiotemporal propagation of genes (see Ref. [54] for an excellent history of the mathematical models of population dynamics), but it can also be used to describe dilute reacting gases [55], nonlinear optics [56], assorted biological population dynamics [57], among others. An example of a PDE with a cubic nonlinearity is the FitzHugh-Nagumo equation

$$\partial_t u = \partial_x^2 u - u(1-u)(a-u), \tag{1.16}$$

which was devised by FitzHugh (originally in ODE form) as a description of spiking behavior in neuronal membrane voltage [58, 59] and experimentally validated using an equivalent circuit by Nagumo *et al.* [60]. This equation can also be used to model excitable systems

such as electronic circuits [61, 62] and cardiac tissue [63].

These types of equation are referred to as reaction-diffusion equations:

$$\partial_t u = D\partial_x^2 u + N(u), \tag{1.17}$$

where $N(u)$ is some nonlinear function describing the reaction process, and D is the diffusion constant which (along with the operator ∇^2) describes spatial coupling. Reaction-diffusion equations can be used to describe combustion processes [64], chemical reactions (where $N(u)$ is often polynomial due to the Law of Mass Action) [65, 66], population dynamics as above, and even more exotic examples like animal coat patterns [67] and criminal behavior [68, 69]. These kinds of equations can then be modified by adding an advection term:

$$\partial_t u = v\partial_x u + D\partial_x^2 u + N(u), \tag{1.18}$$

which can be referred to as an advection-reaction-diffusion equation. By considering advection, this model acknowledges that the flux of u, carried with velocity v, plays a significant role in the dynamics of the variable of interest. These types of equations can describe the dispersal and aggregation of organism populations with directed movement (e.g., due to being immersed in a moving fluid or following a gradient in the environment) [70, 71, 72, 73].

Setting $v = u$ and $N(u) = 0$ yields Burgers equation:

$$\partial_t u = \partial_x^2 u + u\partial_x u, \tag{1.19}$$

which was first introduced by Bateman to illustrate the idea of discontinuous solutions to fluid flows [74] and later revisited by Burgers in an effort to provide a simplified model for hydrodynamic turbulence [75]. The equation can be used to model shock waves in aerodynamics [76] and (with some modifications) plasmas [77, 78], in addition to generally

acting as a a prototype for PDEs with discontinuous solutions. An interesting extension of a fundamental equation is the nonlinear, or cubic, Schrödinger equation

$$i\partial_t u = \frac{1}{2}\partial_x^2 u - k|u^2|u, \tag{1.20}$$

which can be used to describe nonlinear waves in optical fibers [79, 80, 81, 82, 83, 84], Bose-Einstein Condensates [85], and nonlinear deep water waves [86]. A related third order equation is the Korteweg-de Vries equation [87, 88]

$$\partial_t u + \partial_x^3 u - 6u\partial_x u = 0, \tag{1.21}$$

which is a model of one-dimensional nonlinear dispersive non-dissipative waves. For a more exhaustive list of PDEs, and some analytic solutions if available, see Ref. [89].

Although these equations were obtained directly from first principles, derived from a first-principles equation by making simplifying assumptions, or by matching phenomenology, they still demonstrate how ubiquitous PDE models are. Therefore, it is important that a symbolic regression procedure be available to identify them when classical model construction fail, so that proper models are available for spatiotemporally complex phenomena.

CHAPTER 2

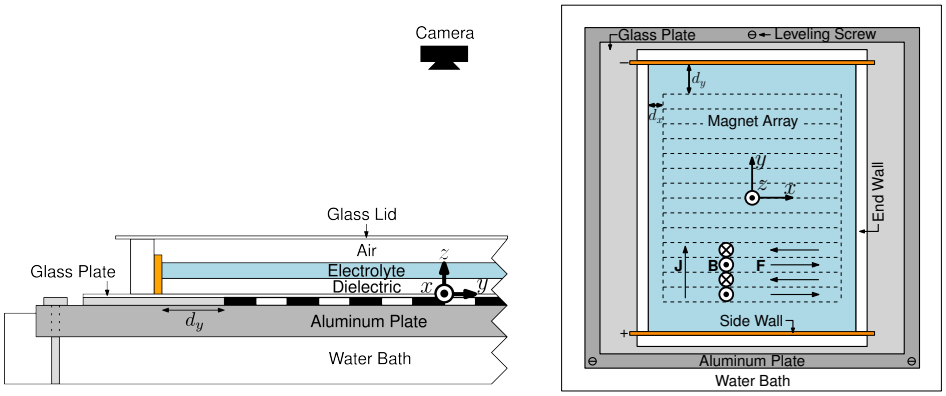
UTILIZING DOMAIN KNOWLEDGE TO CONSTRUCT A LIBRARY

This chapter provides an example of how to build the library $\{F_n\}_n$ by utilizing available physical intuition about the system, sometimes referred to as "domain knowledge." In this chapter and later a thin fluid flow will serve as the illustrative system of choice, because it has a sufficiently complicated model structure, it has latent variables, there is existing physical knowledge about the physics, and there are experimental measurements available for analysis.

2.1 Kolmogorov-like Flow Description

The system comprises a two-immiscible-layer fluid in a rectangular container whose vertical extent (z-direction, thickness $h = 6$ mm) is much smaller than the sides (x and y direction) of its rectangular horizontal area: $L_x \times L_y = 17.78$ cm$\times 22.86$ cm. The main working fluid is an electrolyte (thickness $h_e = 3$ mm, density $\rho_e = 1192$ kg/m^3, viscosity $\nu_e = 5.85$ mPa·s) subject to a uniform current density flowing through the longer of the two horizontal directions (y) due to the influence of opposing electrodes placed in the fluid, such as in Ref. [90]. On the bottom vertical boundary exists a bar magnet array (14 magnets of alternating polarity, each 15.24 cm long and 1.27 cm wide) that erects a nearly oscillatory magnetic field within the fluid domain, which in turn forces the field in a Kolmogorov-like manner. Below the electrolyte is a dielectric (thickness $h_d = 3$ mm, density $\rho_d = 1769$ kg/m^3, viscosity $\nu_d = 1.30$ mPa·s) whose purpose is to dampen the drag effects of the bottom confinement, effectively increasing the two-dimensionality of the dynamics in the electrolyte layer. Above the electrolyte is open to the air.

A diagram of the experimental apparatus is shown in Fig. 2.1. The horizontal motion of the fluid is captured via tracer particles and recorded with the shown camera. The raw



(a) (b)

Figure 2.1: (a) A side cross-section of one of the experimental apprati. (b) A top-down view. The blue shading denotes the fluid domain, the orange bars indicate the electrode locations with charge indicated by $+$ and $-$, the magnetic field alternates between in and out of the field, the current density J flows vertically, and the forcing alternates from left to right vertically with the magnetic field.

videos are then used to extract the velocity data using Particle Image velocimetry (PIV) [91, 92]. Thus, only data for the horizontal velocities at one height are available, and symbolic regression seeks to find a two-dimensional model that effectively describes the spatiotemporal dynamics of the horizontal velocities. It is worth noting that although the evolution of the full three-dimensional velocity is governed by the NSE, a model governing only the horizontal velocities is *not* known. However, with certain assumptions, a two-dimensional model can be derived for the horizontal velocities.

2.2 Deriving a Depth-averaged Model

The fluid contained within the apparatus shown in Fig. 2.1 obeys the incompressible Navier-Stokes equations:

$$\rho\left(\partial_t \mathbf{U} + (\mathbf{U}\cdot\nabla)\mathbf{U}\right) = -\nabla p + \mu\nabla^2\mathbf{U} + \mathbf{F}, \tag{2.1}$$

along with the constraint

$$\nabla\cdot\mathbf{U} = 0, \tag{2.2}$$

where $\mathbf{F} = \mathbf{J} \times \mathbf{B} + \rho\mathbf{g}$ are the external body forces acting on the fluid. The Lorentz force $\mathbf{J} \times \mathbf{B} = J(B_z\hat{x} - B_x\hat{z})$ arises due to the uniform current density $\mathbf{J} = J\hat{y}$ flowing through the electrolyte, and the gravity force $\rho\mathbf{g} = -\rho g\hat{z}$ acts only in the vertical direction. Previously the equation has been simplified by depth-averaging (2.1) after employing the ansatz

$$\mathbf{U} = P(z)\left(u(x,y,t)\hat{x} + v(x,y,t)\hat{y}\right) + w(x,y,z,t)\hat{z}. \tag{2.3}$$

The resulting horizontal, two-dimensional governing equation is

$$\partial_t\mathbf{u} + \beta(\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\varrho}\nabla p + \bar{\nu}\nabla^2\mathbf{u} - (\alpha+\theta)\mathbf{u} + \mathbf{f}, \tag{2.4}$$

where now the two-dimensional gradient is used $\nabla = (\partial_x, \partial_y)$, $\mathbf{u} = (u, v)$, the forcing is given by

$$\mathbf{f}(x,y) = \frac{J}{\varrho}\int_{h_d}^{h} B_z(x,y,z)dz\hat{x} \tag{2.5}$$

and the parameters are given by

$$\varrho = \int_0^h \rho(z)P(z)dz \tag{2.6a}$$

$$\beta = \frac{\int_0^h \rho(z)P^2(z)dz}{\varrho} \tag{2.6b}$$

$$\bar{\nu} = \frac{\int_0^h \mu(z)P(z)dz}{\varrho} \tag{2.6c}$$

$$\alpha = \frac{\int_0^h \mu(z)\partial_z^2 P(z)dz}{\varrho} \tag{2.6d}$$

$$\theta = \frac{\int_0^h \rho(z)w\partial_z P(z)dz}{\varrho} \tag{2.6e}$$

The physical parameters are dependent on the vertical dimensional only as piece-wise constant functions, reflecting the binary setup. The last term includes the vertical velocity,

so further simplification is motivated. Using (2.3) and (2.2), the equivalence

$$w(x,y,z) = \nabla \cdot \mathbf{u}(x,y) \int_0^z P(z')dz' = \nabla \cdot \mathbf{u}(x,y)\mathcal{P}(z) \tag{2.7}$$

is established. This can then be used to refine the definition

$$\theta = \gamma \nabla \cdot \mathbf{u}, \tag{2.8}$$

where

$$\gamma = \beta - \frac{(\rho_d - \rho_e)\mathcal{P}(h_d)P(h_d) + \rho_e \mathcal{P}(h)}{\varrho}, \tag{2.9}$$

has been simplified with the boundary conditions on P: $P(0) = 0$ and $P(h) = 1$. These boundary conditions mean that $\mathbf{u}$ can be interpreted as the horizontal velocity at the electrolyte-air interface. Finally, this yields the governing equation

$$\partial_t \mathbf{u} = -\beta(\mathbf{u} \cdot \nabla)\mathbf{u} + \bar{\nu}\nabla^2 \mathbf{u} - \alpha \mathbf{u} + \gamma(\nabla \cdot \mathbf{u})\mathbf{u} - \frac{1}{\varrho}\nabla p + \mathbf{f}(x,y). \tag{2.10}$$

A governing equation for the pressure can be obtained by noting that the z-component of (2.1) is dominated almost entirely by the balance between the vertical pressure gradient and gravity, then integrating said equation and relating the temporal evolution of the height of the fluid to the vertical velocity given by (2.7). This results in the evolution equation

$$\partial_t p = -\kappa \nabla \cdot \mathbf{u}, \tag{2.11}$$

where

$$\kappa = \frac{g\left((\mathcal{P}(h) - \mathcal{P}(h_d))\rho_e(h_e + h_d) + (\mathcal{P}(h_d) - \mathcal{P}(0)(\rho_e h_e + \rho_d h_d))\right)}{\varrho}. \tag{2.12}$$

Using the physical parameters corresponding to the two fluids, and using the size of

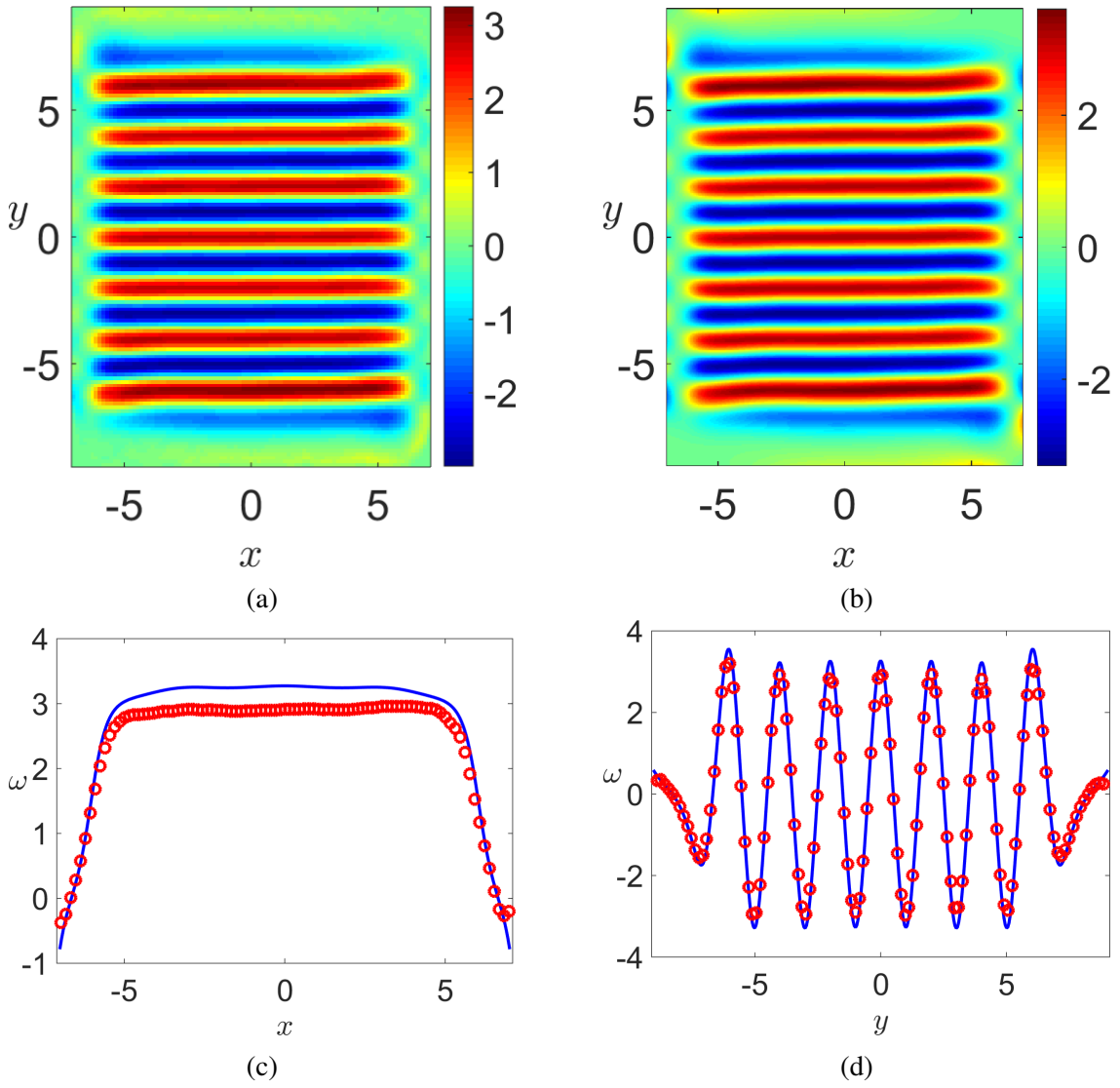


Figure 2.2: Comparison of the vorticity representation of the laminar fixed point for (a) experiment and (b) weakly-compressible simulation. The vorticity is also compared for cross sections in y (c) and x (d) [93].

the apparatus, these depth-averaged parameters can be computed. For example, for one of the apparatus the parameters are $\beta = 0.826$, $\bar{\nu} = 3.2312 \times 10^{-6}$ m^2/s, $\alpha = 0.0645$ s^{-1}, $\gamma = 0.16$, and $\kappa = 0.0549$ m^2/s^2. A more detailed derivation, along with order of magnitude arguments for the exclusion of certain terms can be found in [93]. Note that the parameters shown here are formally dependent on x and y, as each integral depends on horizontally varying fluid depths, but experimentally the height of the free surface varies on the order of μm, in comparison to the average values of $\bar{h}_e \approx \bar{h}_d \approx 3$ mm, so the

spatial variation in the parameters is expected to be minor. Earlier studies [94] considered a horizontally incompressible fluid, which corresponds to setting $\partial_z w \approx 0$, and subsequently $\gamma = 0$. This incompressible version has also been proposed before, such as in Refs. [95, 96, 97].

For low current values (the forcing $\mathbf{f} = (f_0, 0)$ is directly proportional to the current density) the model (2.4) recreates the flow remarkably well. For such low forcing, the base state resembles the forcing profile extremely closely, shown in Fig. 2.2. This close resemblance between model and observation is valid, because the assumption on the flow made in (2.3) is close to being true. In particular, the vertical scaling of the velocity $P(z)$ is computed using rather drastic assumptions on the horizontal velocity: $\mathbf{u} = P(z)\sin(k_0 y)\hat{x}$, where k_0 is the wavenumber corresponding to the frequency of the forcing profile. This drastic assumption on the form of the velocity results in a solvable ODE for $P(z)$, whose solution is then used to compute the parameters in (2.6). This assumption isn't even completely valid at low Re, only in the inner domain is it approximately true. Furthermore, the y-velocity v increases in magnitude at the primary instability, making the calculated parameters become less and less theoretically sound.

The primary instability is in fact characterized by both the emergence of nontrivial v magnitude in the inner domain, as well as a modulation of the horizontal bands indicating x dependence in the inner domain also emerges. Both of these imply that the computed depth-averaged parameters contain errors. However, the modulated state is still computed with qualitative accuracy, shown in 2.3 for the compressible model. It is also computed with good agreement for the incompressible model. However, there exist quantitative differences between simulation and experiment for both the predicted critical Re at which v develops, the average modulation wavelength λ_x, as well as the shape of the bifurcation curve. Modifying the incompressible parameters individually improves agreement for the critical Re and modulation wavelength, implying that a two-dimensional model exists that better fits the data, but said model has different parameters that may vary with forcing

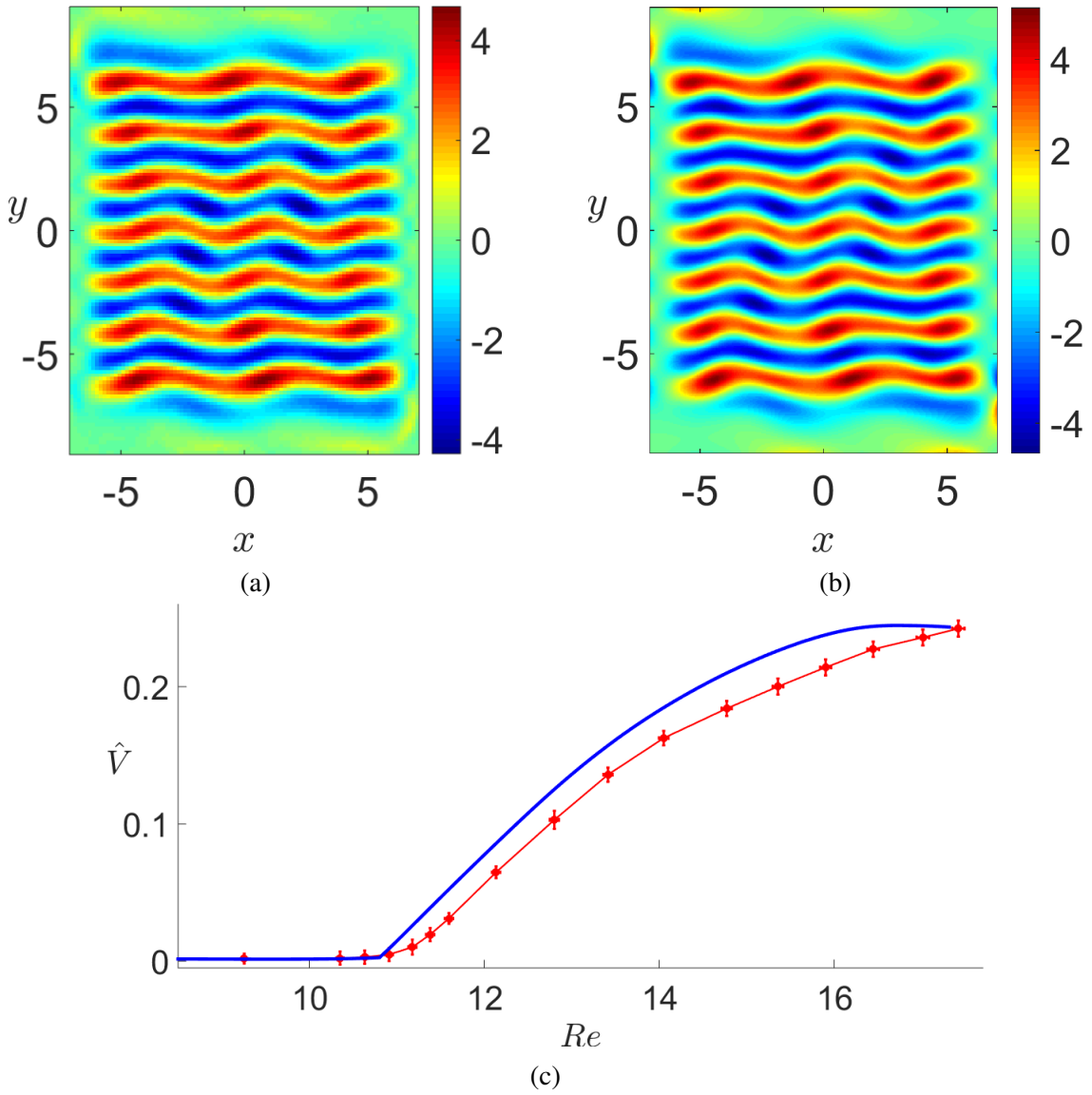


Figure 2.3: The modulated flow, post primary instability is shown for experiment (a) and compressible simulation (b). The order parameter ($\hat{V} = v$) is shown in (c) for the compressible simulation (blue) and experiment (red) [93].

magnitude.

2.3 Exploiting Available Physical Intuition

The model discussed in Section 2.2 shall be generalized here for use within the context of system identification. Since it could be that altogether terms are missing from the model, a generalized system of PDEs needs to be crafted. Although it would be possible to con-

struct a library involving an arbitrary mixture of derivatives and nonlinearities, without a systematic way to constrain the form of the terms in $\{F_n\}_n$, there would be far too many possibilities for symbolic regression to be tractable. To make the problem tractable, the relevant physics and symmetries of the problem can be used to impose a number of constraints on the form of the model and the choice of the dependent variables.

Being a fluid flow, the system can be described by two fields, velocity $\mathbf{u}$ and pressure p. So, the dynamics of the fluid flow should be described by evolution equations of the general form

$$\begin{aligned} \partial_t \mathbf{u} &= \mathbf{N}_u(\mathbf{u}, p), \\ \partial_t p &= N_p(\mathbf{u}, p), \end{aligned} \tag{2.13}$$

where $\mathbf{N}_u$ and N_p are some (generally nonlinear) differential operators. The form of these operators is constrained by the physics and symmetry; first, consider the latter. Since $\partial_t \mathbf{u}$ is a vector, in order to preserve the rotational symmetry, $\mathbf{N}_u$ also has to be a vector. So, $\mathbf{N}_u$ can be constructed as a linear superposition of terms each of which is a vector. Since the fluid layer is thin, the vertical component of the velocity is small compared to the horizontal component, so consider $\mathbf{u}$ to be two-dimensional ($\mathbf{u}$ can be thought of as describing the flow at the free surface of the electrolyte). Furthermore, again due to the small thickness of the fluid layer and the dielectric layer buffering the effect of the bottom boundary, both $\mathbf{u}$ and p can be considered functions of horizontal coordinates x and y and time t, but not the vertical coordinate z.

There are several ways to construct a vector out of $\mathbf{u}$, p, the gradient operator ∇, and the external forcing field $\mathbf{f}$ (assumed to be time-independent). The gravitational acceleration $\mathbf{g}$, the only other vector quantity in the problem, cannot be included in the two-dimensional model, since the latter does not explicitly include the vertical direction. Using one vector object, three vector fields that are linear in $\mathbf{u}$, p, and $\mathbf{f}$ can be constructed: ∇p, $\mathbf{u}$, and

$\mathbf{f}$. More complicated vector fields can be constructed using powers of ∇ and/or nonlinear functions of p, $\mathbf{u}$, and $\mathbf{f}$. Only terms that are linear in p and $\mathbf{f}$ will be considered, since ∇p and $\mathbf{f}$ both describe the (volumetric) force density and they are linearly related to the time rate of change of the momentum density $\rho\partial_t\mathbf{u}$ according to Newton's 2nd law.

More vector fields can be constructed using several copies of $\mathbf{u}$ and ∇. Keeping terms up to third order in $\mathbf{u}$ and second order in ∇, the following evolution equation for the velocity field is obtained

$$\begin{aligned}\partial_t\mathbf{u} = {} & c_1(\mathbf{u}\cdot\nabla)\mathbf{u} + c_2\nabla^2\mathbf{u} + c_3\mathbf{u} + c_4(\nabla\cdot\mathbf{u})\mathbf{u} \\ & + c_5(\nabla\cdot\mathbf{u})^2\mathbf{u} + c_6(\nabla\times\mathbf{u})^2\mathbf{u} + c_7\mathbf{u}^2\mathbf{u} \\ & + c_8\nabla p + c_9\mathbf{f}. \end{aligned} \tag{2.14}$$

The evolution equation for the pressure can be constructed in a similar manner, with N_p that should be a scalar. The pressure should be a function of the velocity only, so keeping the leading order (in ∇ and $\mathbf{u}$) term, we will find

$$\partial_t p = -\kappa\nabla\cdot\mathbf{u}, \tag{2.15}$$

where κ is another unknown parameter. Using the scaling freedom in defining the latent field p explicit in the equations (2.14) and (2.15), without the loss of generality we can set $|c_8| = 1$. Similarly, we can set $c_9 = 1$, which amounts to choosing a particular scale for the (unknown) forcing. The remaining constant (to preserve the translational symmetry in space and time) parameters c_1 through c_7 (and possibly κ) need to be determined from data using sparse regression.

The combination of symmetry and physics constrains the form of the evolution equations rather significantly, yielding a set of evolution equations with few superfluous terms. The last few terms in the general model (2.14) represent vertical flow effects neglected by approximations made during the derivation of the original model in Section 2.2. Specifi-

cally,

$$\alpha = -c_3 - c_4 \nabla \cdot \mathbf{u} - c_5 (\nabla \cdot \mathbf{u})^2 \\ - c_6 (\nabla \times \mathbf{u})^2 - c_7 \mathbf{u}^2 \tag{2.16}$$

can be thought of as a second-order (in $\mathbf{u}$ and ∇) model of the Rayleigh friction coefficient. However, the derived model is very close to the one obtained by depth-averaging.

This chapter serves as an illustration of how traditionally a model is only as good as the assumptions that go into its derivation, and the accuracy of models deteriorate when the assumptions no longer hold. Using much of the same domain knowledge used in the traditional derivation, a more general model was then constructed. In doing so, using this quasi-2D fluid flow as an example, the general process of utilizing physical intuition to build a library was demonstrated.

CHAPTER 3

DIRECT IDENTIFICATION OF QUASI-2D NAVIER STOKES

Now that a library of candidate model terms has been constructed by using domain knowledge, the quasi-2D flow can be further exploited to investigate symbolic regression. This chapter focuses on using the strong form of (2.14), and so a method for evaluating the PDE at local points need be devised. Along with this, a sampling scheme needs to be chosen so that the available information is utilized in an efficient manner. Finally, after the linear system has been constructed, the iterative method used to identify relevant terms is described.

Once the details of the method are shown, the results will be presented. Namely, the questions "how well can the models be identified with and without external noise corruption?" and "for what method parameters yield the optimal results?" will be answered. The scaling with one of the principal factors of the method: the noise amplitude will be given theoretical treatment. Finally, some guidelines as to the limits of how identifiable terms are when they are technically present, but weakly so (such as the weakly compressible term) are also presented.

3.1 Direct Operator Selection

The data characterizing both components of the velocity field $\mathbf{u}$ is assumed to be available on a uniform grid. However, unlike standard sparse regression problems where all the variables are directly observable, in this problem neither the pressure p nor the forcing $\mathbf{f}$ are, so both fields have to be either determined independently or eliminated. In principle, for a fully resolved incompressible ($\kappa \rightarrow \infty$) flow field, if $\mathbf{f}$ were known, p could be obtained in a standard way by applying a divergence to (2.14), which yields a pressure Poisson equation. Typical experimental data however have a resolution that is too poor

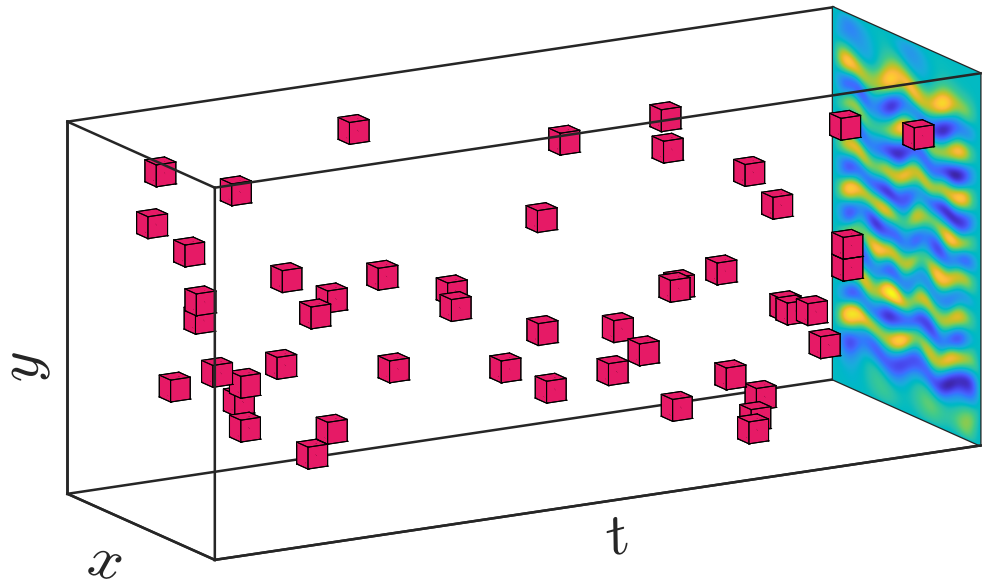


Figure 3.1: Sampling scheme for the direct evaluation of the thin fluid flow. The grey dots represent the (sub-sampled by a factor of 40^2 in space) discrete locations for which the data are available, the red circles correspond to the locations randomly chosen points to locally evaluate the (modified) governing equation, and the green boxes correspond to the neighborhood surrounding the point used to smooth the data and evaluate derivatives. Spatial axes are to scale relative to each other, and the spatial sizes of the fitting neighborhoods. The temporal extent of the available data is too long to show to scale, so a sub-domain is used for illustrative purposes. In the back is an example snapshot at the relevant Re.

(and noise level that is too high) to make it possible to compute pressure in this manner.

In the following, we will focus just on the evolution equation for the velocity field; the evolution equation for the pressure is very simple and the coefficient κ can be eliminated altogether by rescaling c_8 and/or p. The terms involving both latent fields can be eliminated from (2.14) by applying an operator $\hat{P}_k = \hat{S}_k \hat{C} \hat{T}$ composed of three operations: $\hat{C} = \hat{z} \cdot \nabla \times$ removes the dependence on ∇p which is curl-free, $\hat{T} = \partial_t$ removes the dependence on $\mathbf{f}$ which is constant, and the sparsification operator $\hat{S}_k$ samples the original data at discrete space-time points $(x, y, t)_k$ which are uniformly randomly distributed throughout the spatiotemporal domain, as visualized by figure 3.1. The corresponding discretization of the resulting PDE (which is second order in time, third order in space, and fourth order

overall) takes the form of (1.4), where the column entries

$$q_{k0} = \hat{P}_k(\partial_t \mathbf{u} - \mathbf{f} + \nabla p), \qquad q_{k1} = \hat{P}_k(\mathbf{u} \cdot \nabla)\mathbf{u}, \tag{3.1}$$

$$q_{k2} = \hat{P}_k \nabla^2 \mathbf{u}, \qquad q_{k3} = \hat{P}_k \mathbf{u}, \tag{3.2}$$

$$q_{k4} = \hat{P}_k(\nabla \cdot \mathbf{u})\mathbf{u}, \qquad q_{k5} = \hat{P}_k(\nabla \cdot \mathbf{u})^2\mathbf{u}, \tag{3.3}$$

$$q_{k6} = \hat{P}_k(\nabla \times \mathbf{u})^2\mathbf{u}, \qquad q_{k7} = \hat{P}_k \mathbf{u}^2\mathbf{u} \tag{3.4}$$

correspond to different terms in (2.14). Note that $q_{k0} = \hat{P}_k \partial_t \mathbf{u}$, so none of the terms $\mathbf{q}_n$ in fact depend on either p or $\mathbf{f}$. For the number of points in the sample, K, exceeding the number of unknown coefficients, this yields 1.4 as an overdetermined system of linear equations for $\mathbf{c}$, where Q and $\mathbf{q}_0$ can be evaluated using any algorithm sufficiently robust with respect to noise and sparsity of the data. The particular procedure used here is described in the next section.

Sparse regression was performed using a variation of the iterative algorithm for sparse identification of nonlinear dynamical systems (SINDy) [9], which involves computing the solution $\mathbf{c}$ that minimizes the residual

$$\eta = \|\mathbf{q}_0 - Q\mathbf{c}\|_1 \tag{3.5}$$

of the linear system (1.4), followed by a thresholding procedure to remove dynamically irrelevant terms. Note that the library terms $\mathbf{q}_n$ themselves can differ by many orders of magnitude (an example of this is presented below). Since it is the product, $c_n\mathbf{q}_n$, that determines a given term's role in the model, a slightly modified thresholding procedure is employed. The norms of the products $c_n\mathbf{q}_n$ are compared to the residual η: the columns of Q for which $\|c_n\mathbf{q}_n\|_1 < \gamma\eta$ are removed, and the process is repeated until all remaining terms are above the threshold. Here γ is a constant that can be above or below, but is close to, unity. This approach requires no *a priori* knowledge of the system, in contrast to

previous studies [9, 14] that have defined various arbitrary thresholds as stopping conditions for the iterative procedure.

3.2 Polynomial Interpolation

All of the library terms involve spatial and/or temporal derivatives of the velocity field. Using total variation regularization of the data [98] to reduce the influence of noise is both prohibitively expensive in higher dimensions and unnecessary given the sparse nature of the system (1.4). Additionally, although spectral derivative estimates perform better in the noiseless case, their high frequency components are corrupted by noise, diminishing their accuracy severely [10]. Therefore, to accomplish the task of smoothing noisy data and taking numerical derivatives concurrently, a higher-dimensional generalization of the polynomial interpolation in [21] is used instead. At each point chosen by the sparsification operator $\hat{S}_k$, the velocity fields were approximated by a polynomial in x, y, and t fitted to discrete data on a rectangular domain Ω of size $2H_x \times 2H_y \times 2H_t$ centered at a grid point $\mathbf{x}_k = (x, y, t)_k$.

In particular, the x-component of the velocity $u(x, y, t)$ near $(x, y, t)_k$ was approximated as

$$\tilde{u}(x, y, t) = \sum_{l=0}^{L} \sum_{m=0}^{M} \sum_{n'=0}^{N'} U_k^{lmn} \bar{x}^l \bar{y}^m \bar{t}^{n'}, \tag{3.6}$$

where the overbar denotes the shifted and rescaled coordinates in which the domain Ω becomes a cube $\Omega' = [-1, 1] \times [-1, 1] \times [-1, 1]$, e.g.,

$$\bar{x} = \frac{x - x_k}{H_x}, \tag{3.7}$$

etc. The order of the polynomial in each direction should be at least as large as the order of the highest derivative appearing in the model equation (2.14) after the operator $\hat{C}\hat{T}$ is applied, but ultimately is a tunable parameter, with the specific choice to be discussed in more detail in the next section.

The coefficients $U_k^{lmn'}$ were found by minimizing the cost function

$$F = \sum_{(x_i,y_j,t_{k'})\in\Omega} w_{ijk'}(u_{ijk'} - \tilde{u}(x_i, y_j, t_{k'}))^2, \tag{3.8}$$

where $(x_i, y_j, t_{k'})$ are the discrete points at which the data are available, $u_{ijk'}$ are the discretely available velocity data, and $w_{ijk'}$ is a weighting function. This is a standard least squares problem whose solution is given by setting $\partial F/\partial U_k^{qrs} = 0$. This yields a system of $(L+1)(M+1)(N'+1)$ linear equations

$$\langle w\, u\, \bar{x}^q \bar{y}^r \bar{t}^s \rangle_\Omega = \sum_{l,m,n}^{L,M,N} U_k^{lmn} \langle w\, \bar{x}^{l+q} \bar{y}^{m+r} \bar{t}^{n+s} \rangle_\Omega, \tag{3.9}$$

where $\langle\cdot\rangle_\Omega$ denotes the average over the spatiotemporal sub-domains for which the local fits are defined. The weighting function $w_{ijk'}$ was used to bias the accuracy of the approximation toward the central point of the domain Ω (where all of the derivatives are evaluated) and is defined as a Gaussian

$$w_k = \exp\left(-\frac{\bar{x}^2 + \bar{y}^2 + \bar{t}^2}{\lambda^2}\right), \tag{3.10}$$

with the width λ being another tunable parameter of the model (here $\lambda = 0.5$). The same procedure was used to determine the coefficients V_k^{lmn} for the y-component of the velocity $v(x, y, t)$.

After the polynomial coefficients have been determined, the k^{th} row of the library Q and $\mathbf{q}_0$ can be constructed by evaluating the respective derivatives of u and v at $(\bar{x}, \bar{y}, \bar{t}) = (0, 0, 0)$ using (3.6). For instance, the linear term evaluated at $(x, y, t)_k$ is

$$q_{k3} = V_k^{101} - U_k^{011}, \tag{3.11}$$

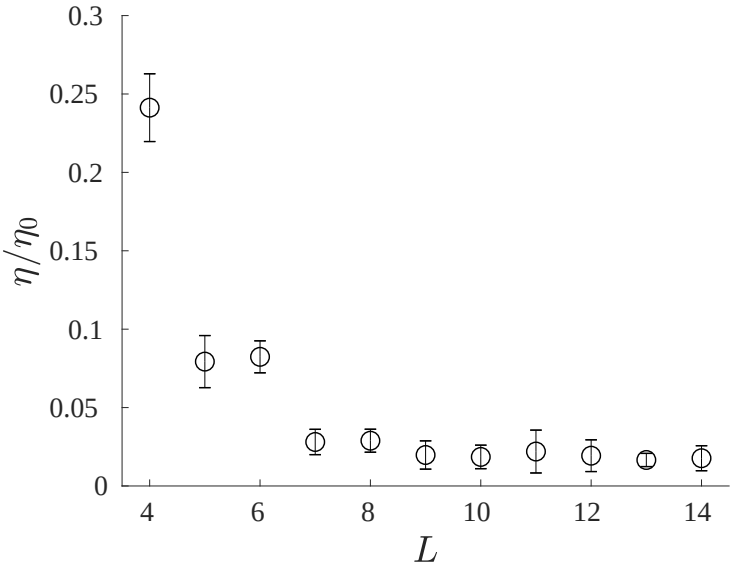


Figure 3.2: Residual as a function of polynomial order for $N = 10$ and $\sigma = 0$. Here, error bars denote standard deviation and symbols denote mean values.

the time-derivative term is

$$q_{k0} = 2V_k^{102} - 2U_k^{012}, \tag{3.12}$$

and the Laplacian term is

$$q_{k2} = 6V_k^{301} + 2V_k^{121} - 2U_k^{211} - 6U_k^{031}. \tag{3.13}$$

The remaining terms are more complicated, and for the purposes of method illustration are omitted. This process was repeated for each point defined by $\hat{S}_k$ in order to completely evaluate the library and the target. Throughout the chapter, $K = 250$ points were used to construct the library; neither the mean nor the standard deviation of the coefficients c_i were found to exhibit meaningful variation for a larger number of points.

3.3 Noiseless Results

Surrogate data used for testing the sparse regression procedure was generated using the model (2.14)-(2.15) with the (non-dimensional) parameters $c_1 = -0.826$, $c_2 = 0.0487$,

$c_3 = -0.157$, $c_4 = 0.164$, $c_5 = c_6 = c_7 = 0$, $c_8 = -1$, and $\kappa = 2015$. To non-dimensionalize the equations, the length-scale used was the width of one of the bar magnets, the velocity-scale was chosen to normalize the forcing magnitude and is proportional to the square root of the driving current, and the time-scale is chosen as the length-scale divided by the velocity-scale. This set of parameters describes a nearly incompressible flow found in the experiment described in Ref. [92], which features the forcing field with a nearly sinusoidal profile in the y direction with period $2\chi = 2$ and amplitude equal to 1.0649 in nondimensional units. The solution describing a weakly turbulent flow was obtained using a numerical integration scheme based on operator splitting as described in Ref. [93]. The linear terms were evolved in time implicitly, while the nonlinear terms were handled via a 2^{nd} order Adams-Bashforth scheme. The solution was integrated on a computational grid with $\Delta x_c = \Delta y_c = 0.025$, and $\Delta t_c \approx 0.02$ (c for "computational"). Gaussian random noise with variance σ was added to both components of the flow velocity $\mathbf{u}$. For reference, the maximal flow velocity is $O(1)$ in non-dimensional units.

In order for the algorithm to produce meaningful results, its various tunable parameters must be properly set. The noiseless case exhibits the least amount of sensitivity to variation of fitting parameters; the only restriction is that the polynomial orders L, M, and N be high enough to capture the variation in the data over Ω. While higher order interpolation allows better approximation of the data, it is also more sensitive to noise. To lessen the influence of noise, a larger number of measurements can be used for each k in the library matrix. There are two ways to achieve this: by increasing the size of the sampling domain Ω or by using a finer grid on which data are measured. The largest size of Ω is effectively limited by the characteristic length and time scales for the problem. In the present problem, the natural length scale is defined by χ. Consequently, here set $H_x = H_y = \chi/2$, such that the width of Ω in both spatial dimensions is equal to $\chi = 1$. There is no natural time scale, so one will be chosen based on the autocorrelation time $\tau \approx 9.9$. In the following set $H_t \approx 0.85\tau$, which is an optimal choice for $\sigma = 10^{-3}$ and $M = L$. Furthermore the finest grid available

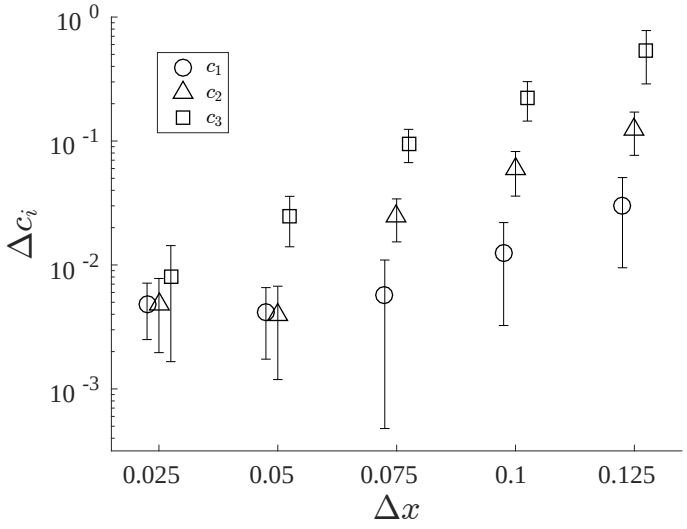


Figure 3.3: The relative error, Δc_i scaling as a function of grid spacing.

in space will be used, i.e., $\Delta x = \Delta x_c$, while in time we use $\Delta t = 25\Delta t_c$. With this choice, Ω corresponds to a $40 \times 40 \times 34$ block of data with dimensions that are roughly comparable in the spatial and temporal directions. Finer grid densities make evaluating the averages in (3.9) computationally expensive.

To investigate how the choice of polynomial order affects the accuracy of the fit and hence the accuracy with which various partial derivatives of $\mathbf{u}$ are evaluated, the residual (3.5) was used. The dependence of η (normalized by the magnitude of the target $\eta_0 = \|\mathbf{q}_0\|_1$) is shown in Fig. 3.2. Here and below, the averages and standard deviation are computed using an ensemble of 40 different realizations of the sampling operator $\hat{S}$. Note that η generally does not vanish even for the noiseless perfect model of the problem due to discretization errors of the numerical solution. Also note that the magnitude of η describes the accuracy with which equation (1.4) is satisfied, not the accuracy of the numerical solution to the model (2.14)-(2.15). As expected, η decreases for low L, but beyond some threshold (in this case $L = 7$), increasing the polynomial order has little effect on the residual. In particular, $L = N = 10$ results in both a low value of the residual and a small error in parameter estimation in the noiseless case, as will be seen below.

Besides the residual η, another measure of accuracy is the relative error in the parame-

ters,

$$\Delta c_n = \left| \frac{c_n - \tilde{c}_n}{c_n} \right|. \tag{3.14}$$

Since the data, whether from simulation or experiment, is acquired on a discrete grid, it is important to quantify the accuracy of the library construction in the context of sparse regression as a function of the grid spacing. The data, created with Δx_c, Δy_c, and Δt_c were sub-sampled such that $\Delta x = \Delta y \approx \Delta t$, and the symbolic regression procedure was performed at each sub-sampling. The resultant relative parameter error is shown in Fig. 3.3. As can be seen, the method is quite sensitive to how refined the mesh is, and unfortunately only reaches $< 1\%$ relative error in all parameters when the grid density is 4 times greater than what is experimentally available.

However, the coefficient c_4 was incorrectly set to zero by the algorithm for all Δx. The failure of sparse regression to correctly identify the value of c_4 can be understood qualitatively by recalling that it is the product $c_4\mathbf{q}_4$ whose magnitude is used to determine whether the corresponding term should be retained or discarded. For this choice of parameters, $\|c_4\mathbf{q}_4\|_1 \lesssim \eta$, suggesting that the magnitude of this term is as small or smaller than the accuracy to which the governing equation can be satisfied. As mentioned previously, the parameter set used here corresponds to a nearly incompressible flow where $\nabla \cdot \mathbf{u}$ is nonzero but very small. Indeed, from (2.14) and (2.15) we find $\mathbf{q}_4 \sim \nabla \cdot \mathbf{u} \sim \kappa^{-1} \approx 5 \times 10^{-4}$. By eliminating the term $c_4(\nabla \cdot \mathbf{u})\mathbf{u}$ representing the effect of compressibility [93] from the model (2.14), sparse regression effectively recognized this fact. The accuracy of the incompressible model has been established previously by comparing experimental and numerical results [94, 99].

To verify that this term was indeed eliminated due to the large value of κ (and not some shortcoming of the method), the analysis was repeated, setting $\kappa = 1$ to amplify the compressibility effects. In this case the term $c_4(\nabla \cdot \mathbf{u})\mathbf{u}$ is retained in the model and

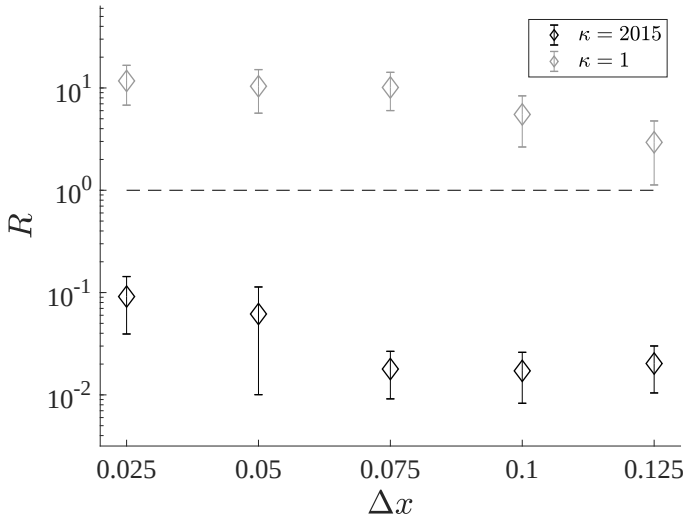


Figure 3.4: The dynamic influence of the compressible term, R_4, evaluated for the weakly-compressible data, for the cases where $\kappa = 1$ and $\kappa \sim \mathcal{O}(10^3)$. The dashed line indicates the strength of the time-derivative term, as a reference. When $R_4 < 1$, the algorithm doesn't identify it as a relevant model term, because although technically there, it isn't a strong influence on the model.

the value of the parameter c_4 is determined correctly for sufficiently small σ. This is a good example illustrating when sparse regression fails to identify terms that are generally required by the physics of the problem but may be neglected under some conditions.

To make the argument more quantitative, introduce the measure

$$R_n = \frac{\|c_n \mathbf{q}_n\|_1}{\eta}, \tag{3.15}$$

of the magnitude of a particular term in the linear equation (1.4) relative to the corresponding residual (3.5). sparse regression can correctly identify a particular term in the model only if the corresponding $R_n > 1$; furthermore, it is expected that the accuracy of parameter reconstruction decreases as R_n approaches unity. For the choice of fitting parameters considered here, R_4 is below unity for $\kappa = 2015$ and above unity ($R_4 \approx 20$) for $\kappa = 1$. This trend is maintained for a range of Δx, as shown in Fig. 3.4. The ratio improves when the grid spacing gets smaller, as η, which decreases with smaller Δx, appears in the denom-

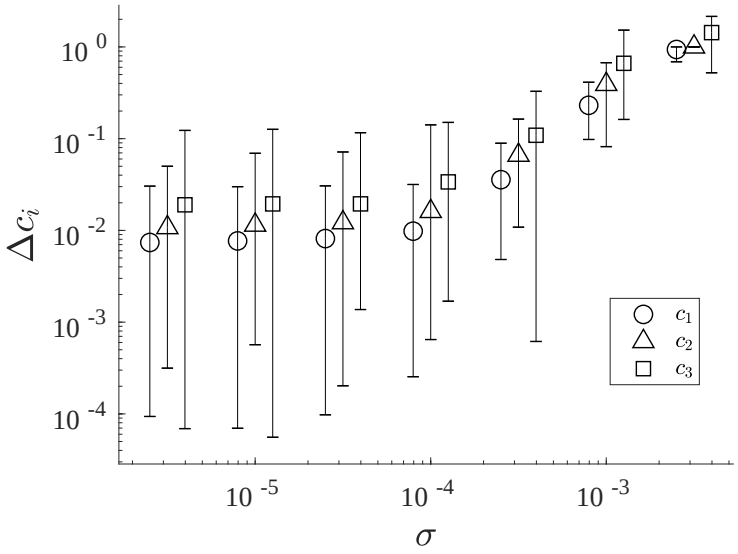


Figure 3.5: Parameter error Δc_i as a function of noise amplitude for $L = N = 10$. The error for c_4 is not shown because SINDy discards the corresponding term. Here and below markers are shifted left or right to avoid overlap. Error bars indicate the full range of data, and markers indicate mean values.

inator in (3.15). Additionally, the terms with lower R_n exhibit the worst fitting accuracy; this explains the larger relative error in c_3 compared with c_1 and c_2 in the $\kappa = 2015$ case (cf. Fig. 3.5), since although $R_3 \approx 30$, it is much smaller than R_1 and R_2, which both have $R_n > 100$.

3.4 Noise Scaling

To determine how the results depend on the amplitude σ of measurement noise, sparse regression was performed and the coefficients $\tilde{c}_n$ produced by the SINDy variation were compared with the reference values c_n used to generate the surrogate data for $\kappa = 2015$. Of the four nonzero parameters used in generating the data, three (c_1, c_2, and c_3) were correctly identified as being nonzero and estimated with a small relative error Δc_n (of order one percent) for sufficiently small σ, as illustrated by Fig. 3.5. Furthermore, the accuracy in estimating all of the remaining parameters decreased sharply for $\sigma \gtrsim 10^{-4}$.

Figure 3.5 also shows that, for the choice of fitting parameters optimized for noiseless

data, the accuracy of sparse regression sharply decreases for $\sigma > O(10^{-4})$. To understand why this happens, define the relative accuracy with which a particular library term is evaluated over the entire sample

$$\xi_n(\sigma) = \frac{\|\mathbf{q}_n(0) - \mathbf{q}_n(\sigma)\|_\infty}{\|\mathbf{q}_n(0)\|_\infty}. \tag{3.16}$$

The effect of noise on the accuracy of all the library terms is shown in Fig. 3.6. Note that the lowest accuracy (highest ξ_n) corresponds to the terms $\mathbf{q}_4$ and $\mathbf{q}_5$ which are linear and quadratic, respectively, in $\nabla \cdot \mathbf{u}$, which is very small. These terms are the most susceptible to corruption by noise but, for large values of κ, they are eliminated by sparse regression anyway. As might be expected, in the absence of these two terms, the term $\mathbf{q}_2$, which involves the highest order derivative (third order in space and first in time), is the least accurate in the presence of noise. This helps explain the difficulties sparse regression has with identifying high order derivatives in all PDE models in the presence of noise. For instance, in a previous study [20], the coefficient of the fourth order derivative term in the Kuramoto-Sivashinsky was determined with a 52% error in the presence of 1% noise. In this case, the terms $\mathbf{q}_3$ and $\mathbf{q}_7$ which involve the lowest order derivative (first in space and time), have the smallest error, suggesting that the order of the derivative is one of the main factors which determine the accuracy of regression in the presence of noise.

The effect of noise can be offset, to some extent by a different choice of parameters. In particular, the order of the polynomial interpolation can be reduced to decrease noise sensitivity. The dependence of the residual η on L is shown in Fig. 3.7 for noise amplitude $\sigma = 10^{-3}$ at which the previous choice of parameters lead to unacceptably large errors. At this value of σ, there is a minimum around $L = 5$ (with a significant increase in η compared to the noiseless case), which represents a balance between the accuracy of the interpolation in capturing the spatial variation of the data at higher L and the noise insensitivity at lower L. In fact, it was found that setting $L = M = N = 6$ is the best choice for minimizing

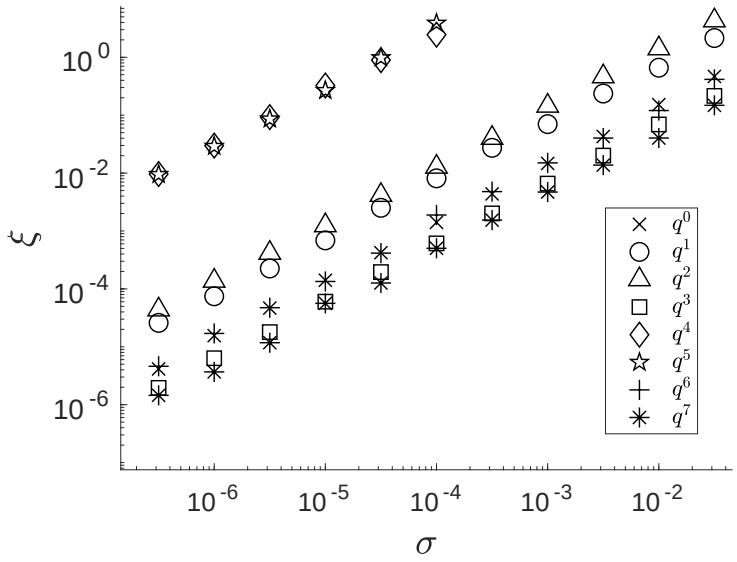


Figure 3.6: Relative accuracy of different library terms as a function of noise amplitude for $L = N = 10$.

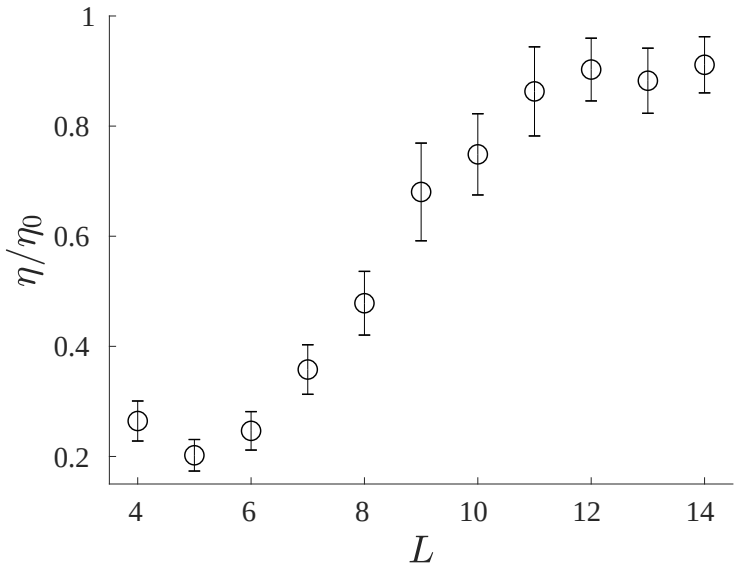


Figure 3.7: Residual as a function of polynomial order for $N = 10$ and $\sigma = 10^{-3}$. Here, error bars denote standard deviation and symbols denote mean values.

both the residual and the error in parameter estimation.

Using the fitting parameters optimized for higher noise levels, sparse regression identifies the correct model (aside from the negligible term $\mathbf{q}_4$) with all the model parameters estimated to within $\sim$10% for 0.3% noise and to within $\sim$30% for 1% noise, as illustrated

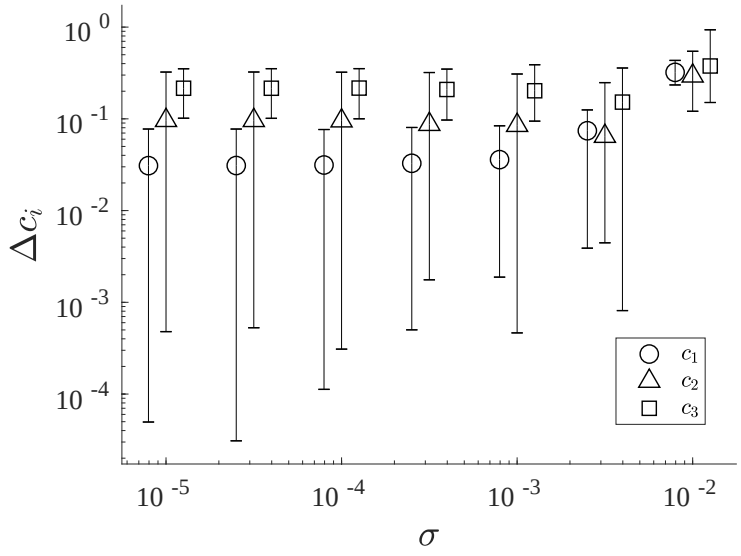


Figure 3.8: Parameter error as a function of the noise amplitude for $L = N = 6$. The error for c_4 is not shown because SINDy discards the corresponding term. Error bars indicate the full range of data, and markers indicate mean values.

by Fig. 3.8. This is comparable to the accuracy achieved for the Kuramoto-Sivashinsky equation [20], which also includes a fourth order derivative. The trade-off of this choice of fitting parameters is the decrease in the accuracy of model parameter estimation at lower noise levels. Furthermore, R_3 becomes close to unity (because η increases so much), so sparse regression yields false negatives for a noticeable fraction of the trials (the data shown in Fig. 3.8 was calculated after discarding the results for which SINDy eliminated the term $\mathbf{q}_3$). In comparison, false negatives did not appear for $L = N = 10$ until fairly high levels of noise. These false negatives occur for lower L and/or N because the magnitude of the residual is determined by the error in the term(s) most affected by the insufficiently accurate approximation (here, the term $\mathbf{q}_2$ which involves the highest order derivative). For the lower L, the variation in the data is not fully resolved, meaning that R_3 is pushed closer to unity (and hence c_3 can be estimated with less accuracy).

Regardless of efforts to offset the influence of noise, there is remarkable universal scaling in Figure 3.6. To understand this scaling, recall that the noise level is the standard deviation of zero-mean Gaussian noise which is added to numerically generated data after

being computed in the absence of noise. So, it must be understood exactly what each of the elements of the library vectors are: combinations of polynomial coefficients computed via a least-squares-error procedure centered around some space-time point $(\mathbf{x}, t)_k$; *ie*

$$q_{kn} = g_n(\mathbf{U}_k, \mathbf{V}_k), \tag{3.17}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the vectors containing the polynomial coefficients for various polynomial orders in x, y, and t, respectively, that approximate the data (and its derivatives), and g_n is how said coefficients are combined to evaluate library term n at point k. For example,

$$g_2(\mathbf{U}_k, \mathbf{V}_k) = -6U^k_{031} - 2U^k_{121} + 6V^k_{301} + 2V^k_{211} \tag{3.18}$$

is the combination corresponding to the Laplacian term; note: g_n becomes nonlinear in the event that $\mathbf{F}_n$ becomes nonlinear.

The polynomial coefficients are determined by solving the linear system obtained by minimizing the standard least squares error metric from above. Arbitrarily ordering the coefficients enables the linear system to be written in matrix form:

$$\hat{A}\mathbf{U}_k = \mathbf{b}_k, \tag{3.19}$$

where the elements of $\hat{A}$ are sums of the polynomials $x_i^{q+l} y_k^{r+m} t_{k'}^{s+n'}$ over the indices for which the discrete data are defined, the vector $\mathbf{U}_k$ contains the polynomial coefficients to be determined arranged in some order, and the vector $\mathbf{b}_k$ contains averages of the observed data weighted by different polynomial powers of the independent variables. Since the ordering can be chosen such that $\hat{A}$ is symmetric, and because $\hat{A}$ is positive definite, $\hat{A}^{-1}$ exists. The positive definiteness of $\hat{A}$ can be shown by noting that since the independent variables are shifted and rescaled such that $x \in [-1, 1]$ on Ω_k (and likewise for y and t), and there are an odd number of discrete grid points centered at (rescaled) coordinates of

$(0,0,0)$, then the elements of $\hat{A}$ are either positive when $q+l$, $r+m$, and $s+n'$ are all even, or 0 otherwise; *ie*

$$\sum_{i,j,k'} x_i^{q+l} y_k^{r+m} t_{k'}^{s+n'} : \begin{cases} > 0, & \text{if } q+l, r+m, s+n' \text{ all even (or 0)} \\ = 0, & \text{otherwise.} \end{cases} \tag{3.20}$$

Hence, the coefficient vector can be written as a linear transformation of the observed data:

$$\mathbf{U}_k = \hat{C}\mathbf{b}_k, \tag{3.21}$$

where $\hat{C} = \hat{A}^{-1}$. Next write the $(L*M*N') \times 1$ vector $\mathbf{b}_k$ containing the observed data as a linear transformation of a $D \times 1$ vector $\mathbf{d}_k$ (where D is the number of discrete elements in the fitting domain Ω_k) containing each individual data measurement:

$$\mathbf{b}_k = \hat{B}\mathbf{d}_k, \tag{3.22}$$

where $\hat{B}$ is the matrix representation of the linear averaging process which is obtained by minimizing the least-squares-error metric. Recall how the observed data is related to the observation noise:

$$\tilde{u}_{ijk'}(\sigma) = u_{ijk'} + \chi(\sigma), \tag{3.23}$$

where $\chi(\sigma) \sim \mathcal{N}(0, \sigma^2)$ and $u_{ijk'}$ is the "true" data value at the point $(x_i, y_j, t_{k'})$. If $\mathbf{U}_k^*$ are coefficients calculated in the absence of noise (*ie* $\tilde{u}_{ijk'} = u_{ijk'}$), then the difference in library elements calculated at zero noise and that for some finite noise amplitude σ is

$$q_{kn}(0) - q_{kn}(\sigma) = g_n(\mathbf{U}_k^*, \mathbf{V}_k^*) - g_n(\mathbf{U}_k(\sigma), \mathbf{V}_k(\sigma)), \tag{3.24}$$

which for linear library terms can be written as

$$\mathbf{G}_n(\mathbf{U}_k^* - \mathbf{U}_k(\sigma)), \tag{3.25}$$

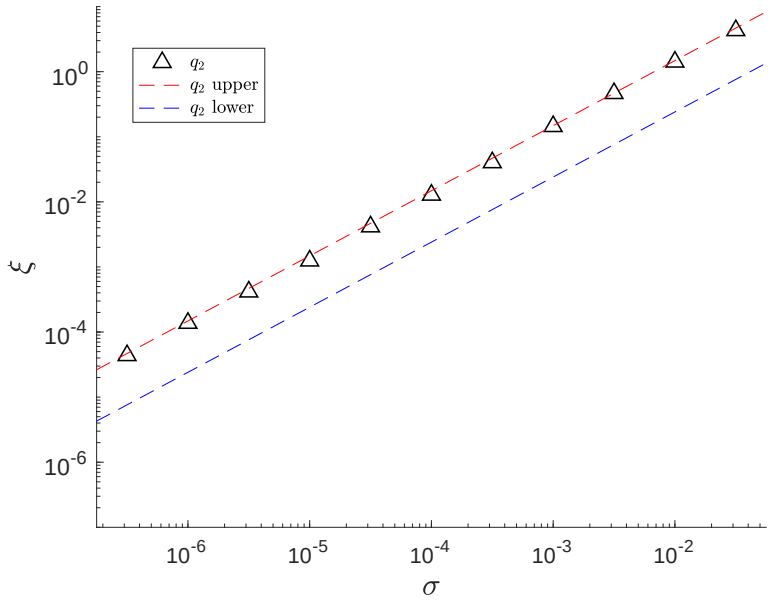


Figure 3.9: An example case ($n = 2$) of the predicted scaling based on the derived bounds on the expectation of $\xi_n(\sigma)$.

where the polynomial coefficients for the two velocities have been combined into one array, and the $1 \times (2 * L * M * N)$ row vector $\mathbf{G}_n$ contains the necessary coefficients to construct the n^{th} library term; for instance, $\mathbf{G}_2$ contains only four non zero elements: 6, 2, -6, and -2, as seen in Eq. (3.18).

Using Eqs. (3.21) and (3.22):

$$q_{kn}(0) - q_{kn}(\sigma) = \mathbf{G}_n(\mathbf{U}_k^* - \mathbf{U}_k(\sigma)) \tag{3.26}$$

$$= \mathbf{G}_n(\hat{C}\hat{B}\mathbf{d}_k^* - \hat{C}\hat{B}\mathbf{d}_k^*) \tag{3.27}$$

$$= \mathbf{G}_n\hat{C}\hat{B}(\mathbf{d}_k^* - \mathbf{d}_k(\sigma)) \tag{3.28}$$

$$= \mathbf{G}_n\hat{C}\hat{B}\mathbf{X}_k(\sigma), \tag{3.29}$$

where $\mathbf{X}_k(\sigma)$ is a vector containing an ensemble of realizations of the random variable $\chi(\sigma)$, since the actual data values disappear when evaluating $\mathbf{d}_k^* - \mathbf{d}_k(\sigma)$. However, this

can be further reduced to the scalar random variable

$$\frac{q_{kn}(0) - q_{kn}(\sigma)}{\|\mathbf{q}_n(0)\|_\infty} = \zeta_{kn}(\sigma) = \sum_{r=1}^{D} p_r \chi_r^k(\sigma), \tag{3.30}$$

which is also normally distributed with $\mathcal{N}(\sum p_r \mu_r, \sum p_r^2 \sigma_r^2) = \mathcal{N}(0, \sigma^2 \sum p_r^2) = \mathcal{N}(0, \tilde{p}^2 \sigma^2)$, since all χ_r are *iid* $\mathcal{N}(0, \sigma^2)$. The coefficients p_r are just a condensed representation of the linear operations $\mathbf{G}_n \hat{C} \hat{B}$ divided by the normalization factor $\|\mathbf{q}_n(0)\|_\infty$. Then the $L - \infty$ norm can be evaluated as

$$\xi_n(\sigma) = \max_k |\zeta_{kn}(\sigma)| . \tag{3.31}$$

The expectation of the distribution of the maximum of an ensemble of K *iid* Gaussian random variables $X_k \sim \mathcal{N}(0, \sigma^2)$ is bounded for any $K \geq 2$ as [100, 101]:

$$0.23\sqrt{\ln K}\sigma \leq \mathbb{E}[\max_{1 \leq k \leq K} X_k] \leq \sqrt{2 \ln K}\sigma, \tag{3.32}$$

and therefore $\xi_n(\sigma) \propto \sigma$ is the predicted scaling when library term n is linear. However, the same scaling is also expected for nonlinear library terms, because to leading order $\xi_n(\sigma)$ is linear in $|\chi| \ll 1$, since $\sigma \ll 1$.

To verify the above claim, the bounds for this case were computed for the Laplacian term $\mathbf{q}_2$:

$$0.23\sqrt{\ln K}\tilde{p}\sigma \leq \mathbb{E}[\xi_2(\sigma)] \leq \sqrt{2 \ln K}\tilde{p}\sigma, \tag{3.33}$$

and the results are shown in Fig. 3.9. As can be seen, the scaling power of 1 is verified, while the measured ξ_2 fall within the bounds of its expectation. Indeed, linearly fitting to the data on the log-log plot gives a scaling power of 0.9972, indicating excellent agreement with the prediction.

With the noise scaling of the method explained, symbolic regression using locally evaluations of the governing equation has been explored here. The method was shown to be capable of identifying the dominant terms from synthetic data, even with state and forcing

measurements being inaccessible. However, the procedure by which this is accomplished is at strong odds with the method's primary detriment: extreme sensitivity to noise corruption. In fact, the noise corruption expected in experiment is $\sim 3\%$ of the velocity magnitude, and so would induce unacceptable parameter estimates from the local evaluation approach. Furthermore, the experimental data exists on a grid too scarce to accurately compute the derivatives, even for clean data. The exploration of these detriments help explain and quantify why and how traditional approaches to symbolic regression for higher-order PDEs fail in the presence of noise and coarsely acquired data, and in doing so motivates searching for alternative solutions. The remainder of this thesis will focus on the development of an alternative solution.

CHAPTER 4

WEAK FORMULATION: INTRODUCTION AND 1D EXAMPLE

The approach considered so far has attempted to utilize local information in order to evaluate derivatives, and it was shown that this is not a valid procedure in the case of higher order derivatives in the presence of noise corruption. Since the small scale information is corrupted by error in the dependent variables, a global procedure is developed below. The following sections are an exposition of the method on a illustrative one-dimensional dynamical system, while more complicated examples are treated in the subsequent chapter. First, the new non-local evaluation procedure is introduced and defined for the Kuramoto-Sivashinsky system; this system also contains a fourth-order derivative that stops the local approach completely. After the method has been introduced, general results of the procedure are shown. Next, estimations for the scaling of parameter error as a function of the method variables are produced. Then, the corresponding numerical experiments are displayed to demonstrate the validity of the scaling estimations. Finally, the method is demonstrated on simple PDEs with non-constant parameters.

4.1 Kuramoto-Sivashinsky and the Integral Operator

The Kuramoto-Sivashinsky equation

$$\partial_t u + u\partial_x u + \partial_x^2 u + \partial_x^4 u = 0, \tag{4.1}$$

describes the chaotic dynamics of laminar flame fronts [102], reaction-diffusion systems [103], and coating flows [104]. This is a notable example of a nonlinear PDE that involves high-order partial derivatives, which has made it difficult to accurately reconstruct from noisy data [20]. Instead of trying to identify the system via direct evaluation, consider

the weak formulation of (4.1). This conversion can be obtained by applying the integral operator

$$\int_{\Omega_i} d\Omega \left[w_j(x,t) \dots \right], \tag{4.2}$$

to the governing equation, where $d\Omega = dxdt$, Ω_i is the integration domain, and $w_j(x,t)$ is some "weight" or "modulating" function that is a function of x and t, but can have different parameterizations indexed by j. The index k on the operator increases with unique combinations of i and j: unique choices of the integration domain and weight function, respectively. The terms in model (4.1) are then

$$q_{k0} = \int_{\Omega_i} w_j \partial_t u d\Omega \tag{4.3}$$

$$q_{k1} = \int_{\Omega_i} w_j u \partial_x u d\Omega \tag{4.4}$$

$$q_{k2} = \int_{\Omega_i} w_j \partial_x^2 u d\Omega \tag{4.5}$$

$$q_{k3} = \int_{\Omega_i} w_j \partial_x^4 u d\Omega, \tag{4.6}$$

where the integration domains are shifted and scaled about some point (x_i, t_i):

$$\Omega_i = \{(x,t) : |x - x_i| \leq H_x, |t - t_i| \leq H_t\}, \tag{4.7}$$

hence the integration domains have widths $F_x = 2H_x$ and $F_t = 2H_t$, in space and time, respectively.

The benefit of the weak formulation becomes apparent when equations (4.3-4.6) are re-written via integration by parts; the time-derivative term is used as an example

$$q_{k0} = \int_{x_i - H_x}^{x_i + H_x} w_j u \big|_{\partial\Omega(x)_i} dx - \int_{\Omega_i} u \partial_t w_j d\Omega. \tag{4.8}$$

Then, if the boundary term is eleminated by judicious choice of the w_j, then the derivatives on the state variable have been completely transferred to the weight function. This then

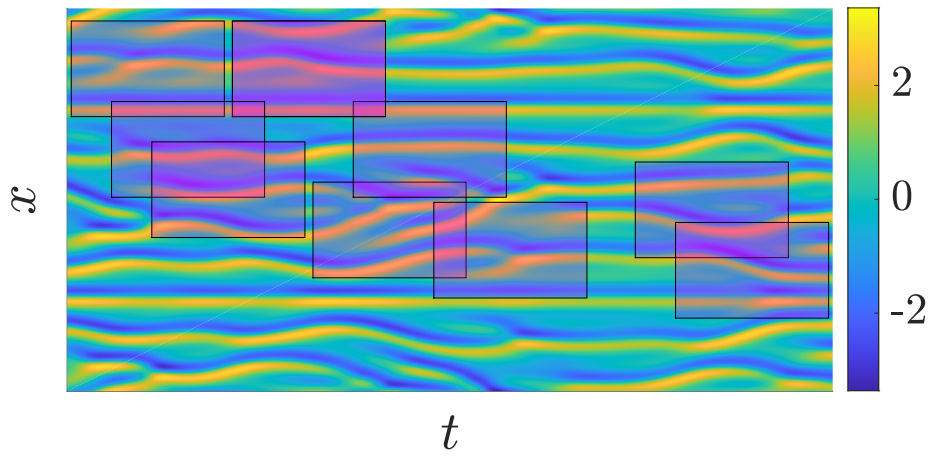


Figure 4.1: Numerical solution to the Kuramoto-Sivashinsky equation with $L_x = 32\pi$ and $L_t = 100$, and an ensemble of integration domains overlaid to illustrate the sampling scheme. The sizes of the integration domains are properly sized with respect to the axes to represent the fractional area of the entire spatiotemporal domain they occupy.

makes the full set of library elements

$$q_{k0} = -\int_{\Omega_i} u\partial_t w_j d\Omega \tag{4.9}$$

$$q_{k1} = -\frac{1}{2}\int_{\Omega_i} u^2\partial_x w_j d\Omega \tag{4.10}$$

$$q_{k2} = \int_{\Omega_i} u\partial_x^2 w_j d\Omega \tag{4.11}$$

$$q_{k3} = \int_{\Omega_i} u\partial_x^4 w_j d\Omega, \tag{4.12}$$

with the boundary conditions

$$w_j(\underline{x}, \pm 1) = 0 \tag{4.13}$$

$$\partial_x^m w_j(\pm 1, \underline{t}) = 0, \text{ for } 0 < m < 3, \tag{4.14}$$

where $\underline{x} = (x - x_k)/H_x$ and $\underline{t} = (y - y_j)/H_t$ are the shifted and re-scaled variables used as arguments for the weight function. Even with the boundary conditions, the weight function

has immense freedom of design. A simple choice is

$$w_{j(l,m)}(\underline{x},\underline{t}) = (1-\underline{x}^2)^\alpha(1-\underline{t}^2)^\beta e^{\pm il\pi\underline{x}}e^{\pm im\pi\underline{t}} = E_\alpha(\underline{x})E_\beta(\underline{t})e^{\pm il\pi\underline{x}}e^{\pm im\pi\underline{t}}, \qquad (4.15)$$

where $E_\delta(s)$ shall be called the "envelope" function. To satisfy the boundary conditions, $\alpha \geq 4$ and $\beta \geq 1$. In principle, any function that is continuously differentiable on $[-1,1]\times[-1,1]$ should be an acceptable choice of weight function, given it is accompanied by the envelope; periodic bases were chosen for the potential ease of interpretation due to the similarity of the integral terms to Fourier coefficients (with the idea that the Fourier spectra of the state variable could help inform the choice of the weight function(s)).

4.2 Kuramoto-Sivashinsky Estimation Results

The linear system (1.4) can now be constructed by evaluating the integrals in (4.12) over a set of randomly chosen domains Ω_i, illustrated by figure 4.1. To test the sparse regression approach, we generated surrogate data by solving the Kuramoto-Sivashinsky equation numerically. To enable direct comparison with the results of Rudy *et al.* [20], the same integrator [105] was used to compute the solution of (4.1) on a spatiotemporal domain of size $L_x = 32\pi$ and $L_t = 100$ using a grid with the same density $\Delta x = 0.0982$ and $\Delta t = 0.4$; an example chaotic solution is shown in Fig. 4.1. Mean-zero Gaussian noise with standard deviation σ was then added to u at each grid point, after which the integrals in (4.12) were evaluated using the trapezoidal rule via MATLAB's built-in `trapz()` function over integration domains with dimensions $H_x \approx 24.5$, $H_t = 20$. The weight function used the exponents $\alpha = 4$ and $\beta = 3$ for its envelope.

The results for different noise levels are shown in Fig. 4.2, with the accuracy of the model reconstruction quantified by the relative errors (3.14), where c_n are the coefficients used to generate the numerical data (all equal to 1) and $\tilde{c}_n$ are the coefficients estimated from noisy data by via our sparse regression algorithm. In the figure, the symbols and

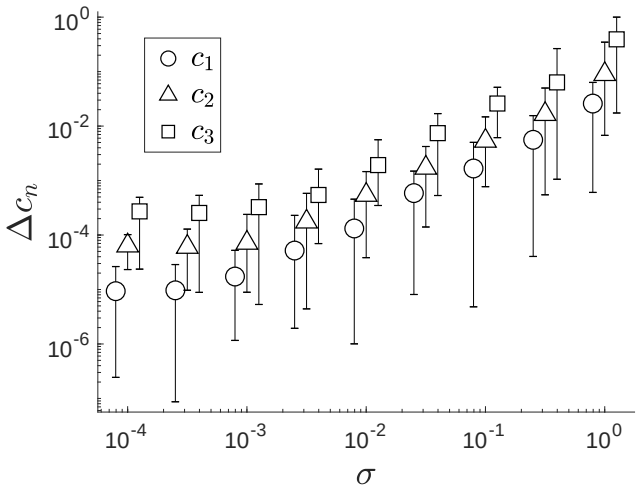


Figure 4.2: The accuracy of parameter reconstruction for the Kuramoto-Sivashinsky equation as a function of the noise amplitude.

the error bars show the mean values and the full range of the results, respectively, for the entire ensemble. Note that the reconstruction remains essentially unaffected by noise, with error of about 1% or below, until the noise level exceeds 10%. This is a dramatic improvement compared to the original study [20], which yielded errors of over 50% for all of the coefficients with just 1% noise.

Quantifying the quality of model reconstruction by comparing the values of the coefficients is simple and straightforward when the reference model used to generate the synthetic data is available. However, even there, a small error in the coefficients can lead to drastically different dynamics, if the system is near a bifurcation. When the data represent experimental measurements, a completely different metric has to be used to quantify the quality of model reconstruction. One natural choice is to compare the predictions of the reconstructed model, after it has been properly initialized, with the observation data. When the dynamics are chaotic, even if the initial condition is specified with exceedingly high accuracy and even if the model is reconstructed perfectly, the predictions will still deviate from observations due to the exponential growth of the initial deviation in time

$$\|\Delta\mathbf{u}(t)\| \sim \|\Delta\mathbf{u}(0)\|e^{\lambda t}, \tag{4.16}$$

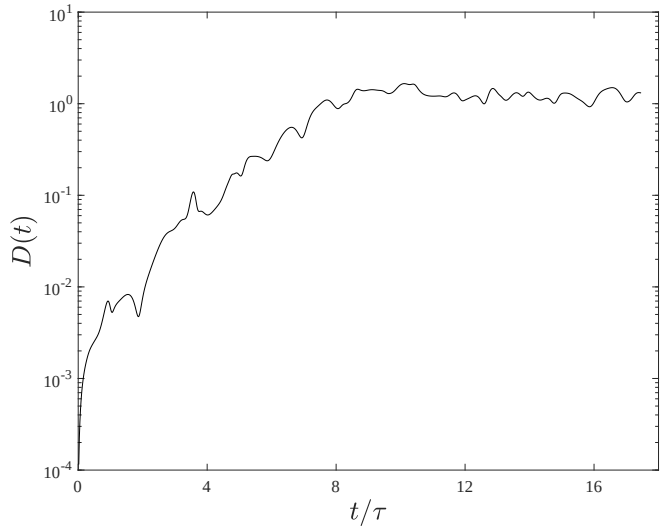


Figure 4.3: Prediction error for the Kuramoto-Sivashinsky equation.

where $\lambda > 0$ is the Lyapunov exponent. Therefore, the reconstructed model can be considered accurate, if it can approximate the observations over a time interval of order $\tau = \lambda^{-1}$. To measure the difference between the observations and model predictions, define the normalized distance

$$D(t) = \frac{\|\mathbf{u}_o(t) - \mathbf{u}_p(t)\|_2}{\langle \|\mathbf{u}_o(t)\|_2 \rangle_t}, \tag{4.17}$$

where $\mathbf{u}_o(t)$ describes the original (synthetic) data, without any noise added, and $\mathbf{u}_p(t)$ is the state predicted by the reconstructed model in the presence of noise, where initially $\mathbf{u}_p(0) = \mathbf{u}_o(0)$. For the model from synthetic data with 1% noise, the approach yielded the following parameters: $c_1 = -1.0001$, $c_2 = -0.9996$, and $c_3 = -0.9987$ (parameter error of 0.13% or less). For the Kuramoto-Sivashinsky equation on a domain of length $L_x = 32\pi$, the Lyapunov time is $\tau = 11.46$, and the prediction error is shown in Fig. 4.3. The model yields accurate predictions (i.e., $D(t) < 0.01$) over a temporal interval of length 2.08τ, which is in line with what can be expected for a perfect model.

4.3 Estimates for Parameter Error Scaling

Although the results are already such that the method drastically outperforms local approaches, the attention now turns to estimating the scaling of the parameter error with the various parameters of the problem: grid resolution, integration domain size, full physical domain size, number of integration domains, magnitude of noise corruption, choice of weight function, and envelope exponents. Because this is a large space over which to optimize the method, in general the method parameters will be varied independently to illustrate the scaling of a single method variable. By understanding the various ways in which the method depends on its parameters, it can be tuned to produce optimal results (ie, with minimal error in estimated parameters).

There are two primary regimes that produce distinct relative error scaling relations. The first corresponds to large amounts of noise corruption. If the noise is some iid random variable X_i, then the effect of the noise should go as an averaged effect of the noise: $S_n = (X_1 + X_2 + \cdots + X_n)/n$. Applying the central limit theorem, S_n should be normally distributed with zero mean and standard deviation $\sigma_X/\sqrt{n}$ (assuming $E[X_i] = 0$). Hence, the relative error should scale as

$$\epsilon_n \sim \sigma\sqrt{\frac{\Delta x \Delta t}{F_x F_t}}, \tag{4.18}$$

where Δx and Δt are the grid spacing of the data when the numerical integrals are estimated (as opposed to the resolution at which the synthetic data was generated), and the fraction in the radical is the inverse of the number of grid points in the integration domain.

On the other hand, when the noise is small, the accuracy of the estimated parameters is controlled by the numerical error. This numerical error in turn can have two different sources. When Δx and Δt are small, the numerical error is dominated by the error of the simulation: the finite accuracy at which the integrator "solves" the Kuramoto-Sivashinsky equation. If the data came from an experiment, this type of error would correspond to

systematic error. For larger Δx and Δt, the numerical error is dominated by the error in computing the integrals. Using numerical quadratures to evaluate the integral

$$I = \int_0^L g(x)dx \tag{4.19}$$

where $g(x) \in C^m$ (i.e., has m continuous derivatives) and $g^{(i)}(0) = g^{(i)}(L)\ \forall i \in [0, m)$ (i.e., all derivatives up to order $m-1$ are periodic on $x \in [0, L]$). Then, for the composite trapezoidal rule on a grid with spacing h, the relative error associated with the discretization on the integration domain can be estimated using exact Euler-Maclaurin formulas [106] and is found to scale as $h^{m+2}|g^{(m+2)}|$ for m even (or $h^{m+1}|g^{(m+1)}|$ for m odd), where a characteristic value of the derivative on $[0, L]$ is used.

This one-dimensional result can be generalized to two dimensions (along with the assumption that the grid spacing is much smaller than the integration domain and correlation length, in space (l_x) and time (l_y)). The relative error due to discretization of the integral of an element of the library matrix Q that involves a temporal derivative of order ν_t or spatial derivative of order ν_x:

$$\epsilon_d \sim \begin{cases} h^{\mu+2},\ \mu \text{ even} \\ h^{\mu+1},\ \mu \text{ odd} \end{cases} \tag{4.20}$$

where $h = \Delta t/l_t \approx \Delta x/l_x$, and $\mu = \min(\alpha - \nu_x, \beta - \nu_t)$. The order of derivatives ranges from 1 to 4, so μ ranges from $\alpha - 1$ to $\alpha - 4$. The extremes in this range can be expected to dominate the error at different values of h: at low h the scaling should be

$$\epsilon_d \sim h^{\alpha-2} \tag{4.21}$$

and at higher h the scaling should be

$$\epsilon_d \sim h^{\alpha} \tag{4.22}$$

assuming that $\alpha = \beta \geq 4$ is chosen to be even.

To connect the relative error estimates of library elements in the two regimes to the parameter error, perturbation theory can be used. First, let $\bar{Q}$ be a matrix containing the exact evaluations of the integrals given in (4.10-4.12) and $\bar{q}_0$ be the the same for the integrals defined by (4.9). Then $\bar{Q}\mathbf{c} = \bar{q}_0$ exactly. However, if there is some error in the library elements, either due to noise or discretization, then the library matrix will be

$$Q = \bar{Q} + \epsilon\tilde{Q} \tag{4.23}$$

where $\epsilon = \max(\epsilon_d, \epsilon_n)$, and $\tilde{Q}$ can be assumed to be a matrix whose entries are distributed as white Gaussian noise and are $\mathcal{O}(F_x F_t)$. On the other hand, the entries of $\bar{Q}$ have a more complicated scaling that depends crucially on the spectral properties of the data, u. To leading order in ϵ, the least squares solution to $Qc = \mathbf{q}_0$ is

$$\mathbf{c} = \bar{\mathbf{c}} - \epsilon\bar{Q}^{+}\tilde{Q}\bar{\mathbf{c}}, \tag{4.24}$$

where $\bar{Q}^{+}$ is the Moore-Penrose pseudo-inverse of $\bar{Q}$. Since the elements of $\tilde{Q}$ can be considered to be uncorrelated,

$$\Delta c_n \propto \epsilon \frac{F_x F_t K^{-1/2}}{\psi(F_x, F_t, l, m)}, \tag{4.25}$$

where the numerator and denominator describe the scalings of $\tilde{Q}$ and $\bar{Q}$'s entries, respectively. Here $\psi(F_x, F_t, l, m)$ encodes the general scaling of an element of the library q_{kn}, with respect to the arguments. Hence, to minimize the parameter error, it is necessary to choose (F_x, F_t, l, m) that maximize ψ.

To begin to understand the scaling encoded in ψ, define an integral

$$I_\alpha(k \mid l, H) \coloneqq \int_{-H}^{H} \left(1 - \frac{x^2}{H^2}\right)^{\alpha} e^{-i\frac{l\pi}{H}x} e^{ikx} dx. \tag{4.26}$$

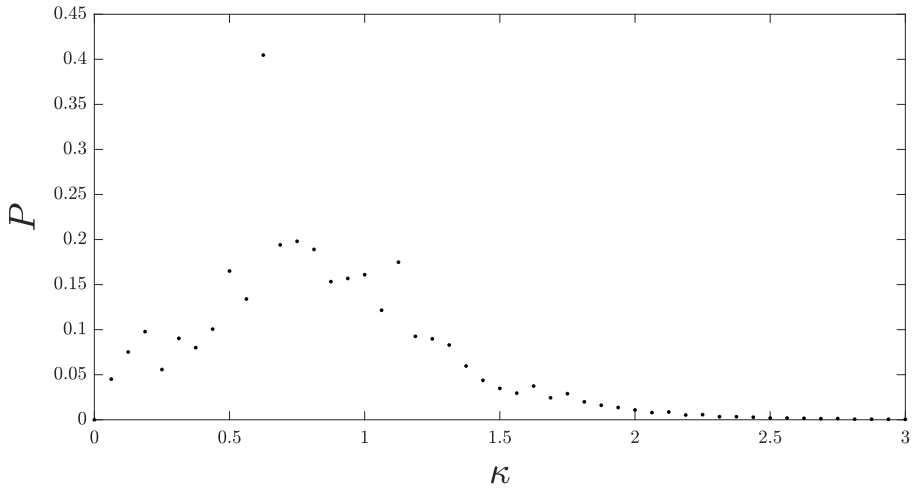


Figure 4.4: Time-averaged spatial Fourier spectra for Kuramoto-Sivashinsky data, in terms of the wave number $\kappa = 2\pi\lambda/L_x$. The spectra is strongly peaked at $\kappa = 1/\sqrt{2}$, which corresponds to wavelength $\lambda = 2\pi\sqrt{2}$, which arises due to a balance between the "hyper-viscous" fourth-order term and the second order term [107].

The functional form of this integral goes as

$$I_\alpha = \frac{P_1(\xi|\alpha)\xi\cos(\xi) + P_2(\xi|\alpha)\sin(\xi)}{\xi^{2\alpha+1}} \tag{4.27}$$

where $\xi = l\pi - kH$, and P_1 and P_2 are some polynomials in ξ such that the Taylor expansion of the numerator about $\xi = 0$ has terms of order $\leq 2\alpha + 1$ equal to 0. The shape of I_α is very similar to the sinc function (and is in fact the sinc function for $\alpha = 0$), but the denominator's exponent changes as a function of α, which in term determines how strongly the wavenumbers outside the main lobe decay. Next, note that for Kuramoto-Sivashinsky data generated on a periodic domain, the state at any given time can be represented as

$$u(x,t) = \sum_{\lambda=-N_x/2}^{N_x/2} U_\lambda(t) e^{i\frac{2\pi\lambda}{L_x}x}, \text{ for } x \in (-L_x/2, L_x/2] \tag{4.28}$$

where L_x is the entire physical domain, and $N_x + 1$ is the number of discretization points in space. The time-averaged spatial spectra for the trajectory used here is shown in figure

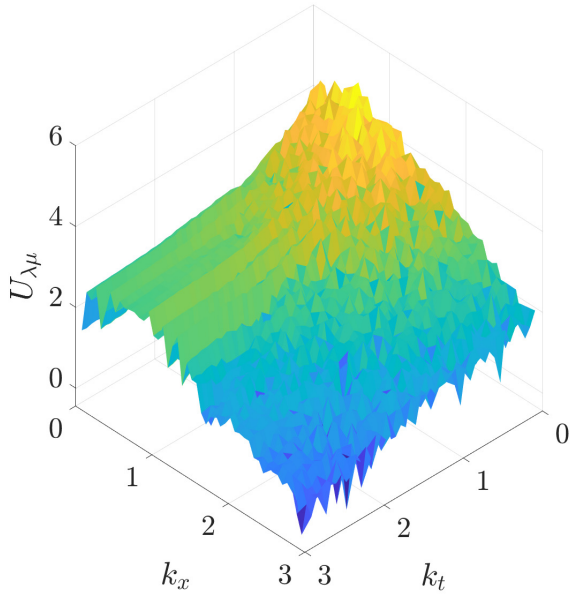


Figure 4.5: The two-dimensional Fourier spectra of the (noise-less) trajectory used to test the scaling. Note that $k_x = 2\pi\lambda/L_x$ and $k_t = 2\pi\mu/L_t$.

4.4. Substituting (4.28) in (4.26) for e^{ikx}, the integral takes the form

$$\begin{aligned}\tilde{I}(\alpha) &= \int_{-H_x}^{H_x} e^{-il\pi x/H_x}\left(1-\frac{x^2}{H_x^2}\right)^{\alpha}\sum_{\lambda} U_\lambda e^{i2\pi\lambda/L_x}dx \\ &= \sum_{n} U_\lambda I_\alpha\left(\frac{2\pi\lambda}{L_x}\,|\,l, H_x\right). \end{aligned} \tag{4.29}$$

The validity of this expression for evaluating the integral (which utilizes the global Fourier spectra to evaluate a local integral) was verified by comparison with the local integral computed via trapezoidal rule as well as the integral calculated using the 0 mode of Fourier spectra acquired on the local domain.

Of course, the library involves both derivatives in space *and* time, and the integrands involve derivatives and nonlinearities of $u(x,t)$. Taking this into consideration, the library

terms represented in terms of the sinc-like functions are given

$$q_{k0} = \sum_{\lambda,\mu} U_{\lambda\mu} I_\alpha(\frac{2\pi\lambda}{L_x}|l, H_x) \left(\frac{i2\mu\pi}{L_t}\right) I_\beta(\frac{2\pi\mu}{L_t}|m, H_t) \tag{4.30}$$

$$q_{k1} = \sum_{\lambda,\mu} U^{(2)}_{\lambda\mu} \left(\frac{i2\lambda\pi}{L_x}\right) I_\alpha(\frac{2\pi\lambda}{L_x}|l, H_x) I_\beta(\frac{2\pi\mu}{L_t}|m, H_t) \tag{4.31}$$

$$q_{k2} = \sum_{\lambda,\mu} U_{\lambda\mu} \left(\frac{i2\lambda\pi}{L_x}\right)^2 I_\alpha(\frac{2\pi\lambda}{L_x}|l, H_x) I_\beta(\frac{2\pi\mu}{L_t}|m, H_t) \tag{4.32}$$

$$q_{k3} = \sum_{\lambda,\mu} U_{\lambda\mu} \left(\frac{i2\lambda\pi}{L_x}\right)^4 I_\alpha(\frac{2\pi\lambda}{L_x}|l, H_x) I_\beta(\frac{2\pi\mu}{L_t}|m, H_t), \tag{4.33}$$

where $U^{(2)}_{\lambda\mu}$ is the Fourier spectra of u^2, $\lambda = 0, \pm1, \pm2, \ldots, N_x/2-1$, $\mu = 0, \pm1, \pm2, \ldots, N_t/2-1$, and the dependence on the location of the integration domain (in combination with the harmonic pair (l, m)) is implied through the subscript on q_{kn}. These expressions' maxima occur when the data's Fourier spectra peak is aligned with that of the sinc-like functions (modified by derivatives). The full two-dimensional Fourier spectra of the data is shown in Fig. 4.5. The appearance of $U^{(2)}_{\lambda\mu}$ is similar to $U_{\lambda\mu}$, but higher modes are more populated due to the nonlinear mixing.

These spectra need to be matched with the filter terms they are paired with in order to minimize the parameter error. Because the data has relatively separable behavior in k_x and k_t, to start m is set to 0 and $H_t = 20$ to reduce the dimensionality of the search for optimality; it then remains only to find the best choice of (l, H_x) that maximize the q_{kn}. At first consideration, it seems intuitive to choose the wave-number of the weight function, $\kappa = l\pi/H_x$, to match the largest wave number of the data $k \approx 0.625$. However, viewing the "filtering" functions,

$$Fi(k, \kappa|\, a(n)) = \left(\frac{2\pi\lambda}{L_x}\right)^a I_\alpha(\frac{2\pi\lambda}{L_x}|l, H_x), \tag{4.34}$$

where $a = 0, 1, 2, 4$ changes based on the library function, as a function of the data wavenumber k and the weight function wave number κ reveals a more complicated story.

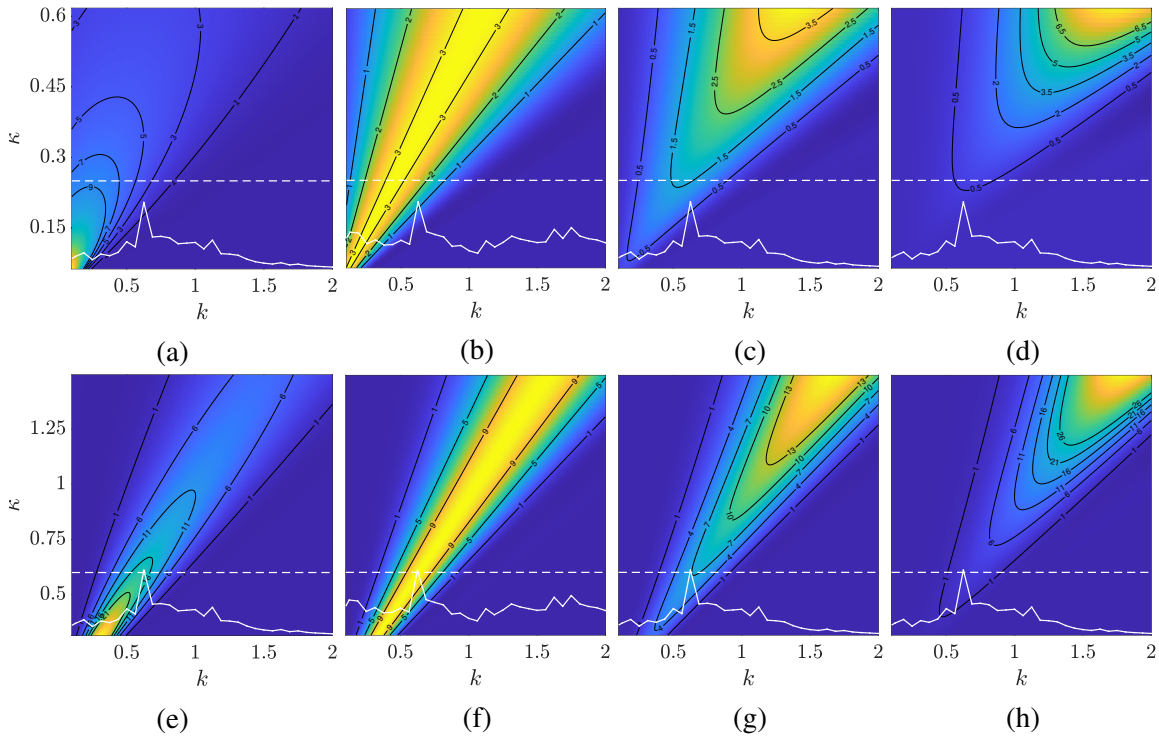
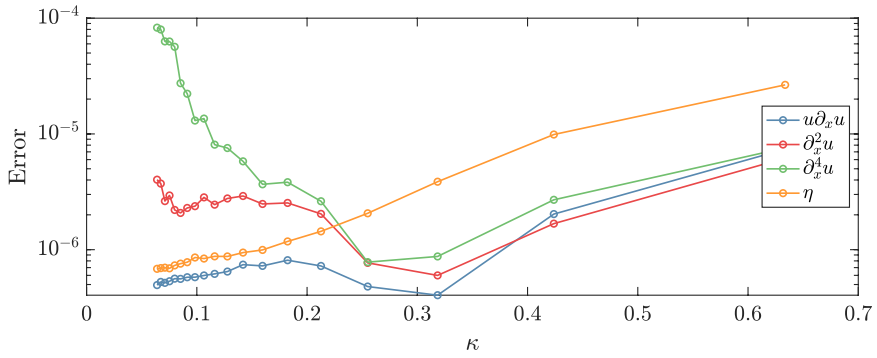


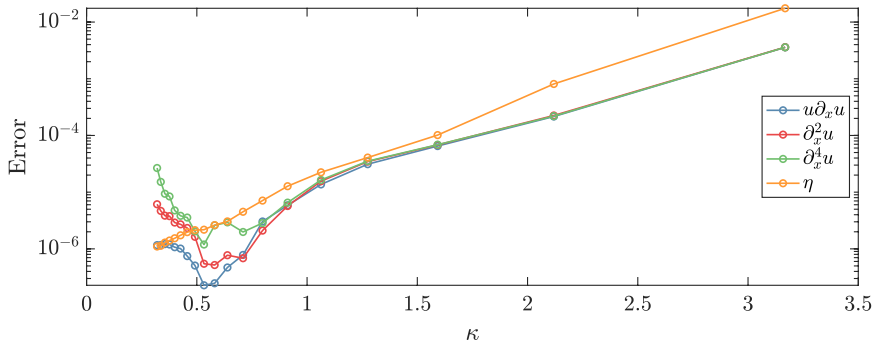
Figure 4.6: The spectral filter functions $Fi(k, \kappa|a)$ for all for library functions for $l = 1$ (top) and $l = 5$ (bottom). Shown in white solid lines are the time-averaged spatial Fourier spectrum for which a κ-slice would be multiplied and summed over λ. The dashed white line indicates the κ at which, for the given l, the minimum in $\max_n(\Delta c_n) = \Delta c_3$ occurs. The library terms increase in n from left to right.

The addition of the derivatives in k space make it so that Fi is significantly different for different library terms. This is shown in Fig. 4.6, along with overlays of the time-averaged spatial Fourier spectra to give an idea of how well the cross section at a given κ would align with $\langle U_\lambda \rangle_t$. This competition implies that the maximal ψ occurs at a κ that offers some compromise between all the library terms.

To test whether or not this is true, κ was used as an independent variable by fixing l and varying F_x (with $m = 0$ and $F_t = 20$). The method was then applied 30 times per κ as a numerical experiment. The results in Fig. 4.7 suggest that this idea of mutual compromise between the library term magnitudes is correct. The κ corresponding to $\min_{n,\kappa}(\Delta c_n) = \min_\kappa(\Delta c_4)$, which can be observed in Figs. 4.7, is shown in dashed white in Figs. 4.6. This dash white line should be used, in combination with the overlaid spectra, to get an



(a)



(b)

Figure 4.7: The parameter error scaling as a function of κ, by fixing $l = 1$ (top) or $l = 5$ (bottom) and varying F_x.

idea of how each term overlaps with the cross-section of the filter function at that κ. For large κ in Fig. 4.6, the filter functions for library terms with higher derivatives have larger magnitude, whereas the time-derivative term is strongly peaked for small κ. The spectrum of the data is peaked somewhere inbetween these two extremes, as are terms with lower-order spatial derivatives. Though this trend is observed for both $l = 1$ and $l = 5$, that the F_i are different functions of κ for different choices of l explain why the minimum in parameter error occurs at different choices of H_x, even though the weight function wave number κ is the same.

Unfortunately, this procedure does not give a systematic way to explain the scaling with respect to the mode of the weight function, the size of the integration domain, or their combined variable: the wavenumber of the weight function. However, generally the optimal wave number of the weight function usually occurs in the vicinity of the peak of

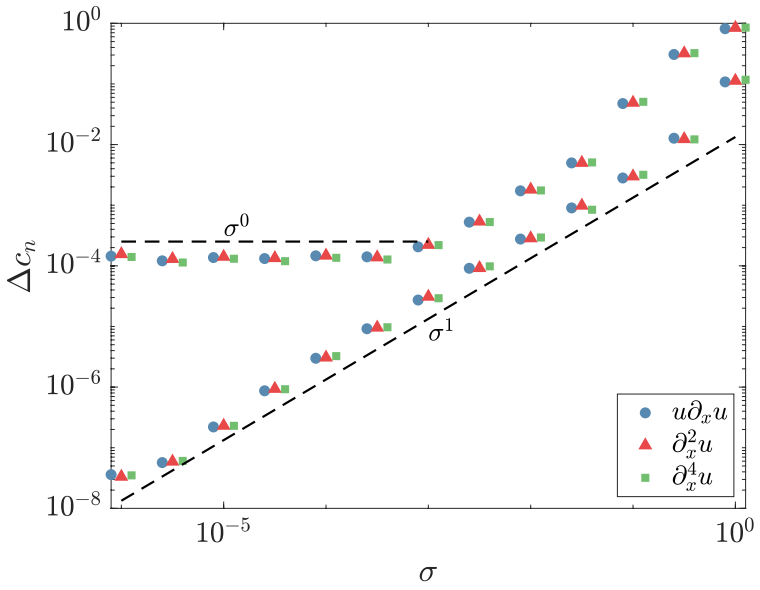


Figure 4.8: Parameter error as a function of additive noise. Method parameters: $\alpha = \beta = 6$, $F_x = 15$, $F_t = 75$, $l = 2$, $m = 1$, and $K = 50$. Markers indicate mean value of 100 runs distinguished by unique sets of integration domain locations.

the spectra of the data.

Putting the issue of optimal integration domain size and/or weight function wavenumber aside, the remaining method parameters are left to be investigated. For low σ, $\epsilon = \epsilon_d$, and therefore Δc_n is independent of σ. For high σ, $\epsilon = \epsilon_n$, so combining (4.25) and (4.18):

$$\Delta c_n \propto \sigma \sqrt{\frac{\delta_x \delta_y}{K F_x F_t}} \frac{F_x F_t}{\psi(F_x, F_t, l, m)}. \tag{4.35}$$

The predicted scaling with respect to σ, in both regimes is consistent with the results shown in Fig. 4.8. This also explains why the parameter error exhibited in the original investigation into the noise scaling in Fig. 4.2 leveled off and showed no further improvement for very low noise. If the grid resolution is increased sufficiently, as in Fig. 4.8, the parameter estimates improve as expected. On the other hand, changing the grid resolution at high noise is quite minor, where $\Delta c_n \propto h$ according to (4.35).

As shown, at low σ the effect of changing h is much stronger; for $\alpha = \beta = 8$, the scaling is $\Delta c_n \propto h^6$ according to (4.20). This dependence is further confirmed by Fig. 4.9, which shows noiseless data. The figure displays the scaling results for two different

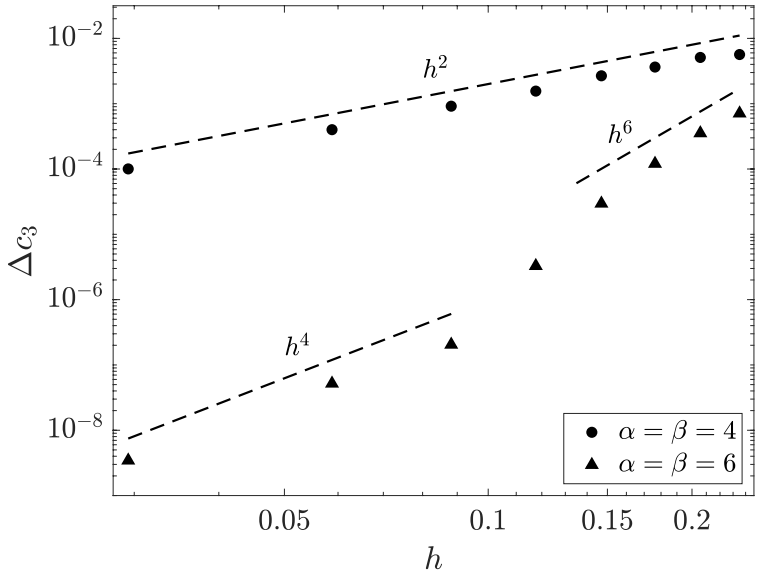


Figure 4.9: Parameter error for the biharmonic term c_4 as a function of $h = \delta_x/l_x \approx \delta_t/l_t$. Two sets of envelope exponents are shown, and the predicted scaling is overlaid as dashed lines.

choices of envelope exponents, to properly test the prediction varies as described not only for h but also for α, β. For the set with $\alpha = \beta = 4$, the scaling is dominated by the term with the fourth derivative, as its scaling h^2 is represented for the entire range of h presented. Alternatively, the dominance is shared between $q_{k3} \sim h^4$ at low h and the rest $q_{k0}, q_{k0}, q_{k0} \sim h^6$ at higher h. In any case, increasing the envelope exponent is a method by which to counteract the deficiency in parameter error induced by an under-resolved grid.

The next important quantity whose effect on the method is important to understand is the number of integration domains used to construct the library matrix K. In general, increasing the number of integration domains used in the library represents using more of the available data. However, intuitively one knows that a given data set is not infinite in time. Thus, eventually the method should receive no more benefit from increasing the number of integration domains. This is what is observed for larger K in Fig. 4.10. On the other hand, there is fairly clear inverse square root scaling before the information in the data set is exhausted. To understand this, return to (4.35), where $K^{-1/2}$ is present clearly. Hence, it is true that using more of the available information to create the library matrix

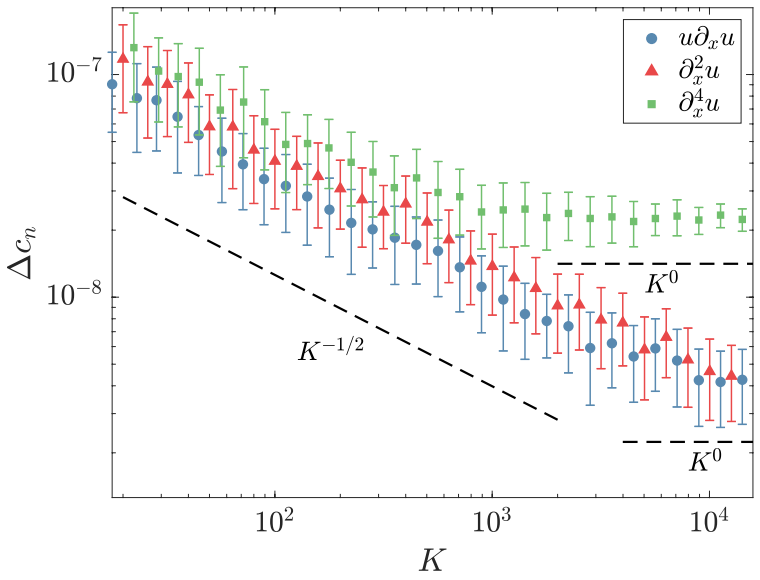


Figure 4.10: Parameter error as a function of number of library rows K. Method parameters: $\alpha = \beta = 6$, $F_x = 15$, $F_t = 75$, $L_x = 16\pi$, $l = 1$, $m = 2$. Markers indicate mean values of 100 runs; errorbars correspond to standard deviation across said runs.

will improve the results. An interesting feature of the figure is that the results change with n. A possible explanation of this could be that the individual terms have less variation, as far as unique information is concerned, across the trajectory. As such, after a certain amount of sampling, no new information is observed.

Similar to the amount of the information used in a given dataset, represented in K, instead datasets of different sizes can be inspected to see if increasing the availabe information beyond one particular type of data set could improve the estimate. To do this, define an area ratio:

$$R_A \coloneqq \frac{L_x L_t}{F_x F_t} \tag{4.36}$$

which is how many integration domains can fit into the entire trajectory area. Fix F_x, F_t and vary L_x, L_t. This is a measure of how much unique information is contained in a given trajectory, and consulting (4.35) once again suggests that the parameter error should scale as $R_A^{-1/2}$ for the same reason as it should with K. To realize a numerical experiment to test this, two different procedures were executed to vary R_A. For both, a long trajectory was generated at $L_x = 32\pi$ and $L_t = 250$. In the first case, $L_t = 250$ is set and L_x is varied; in

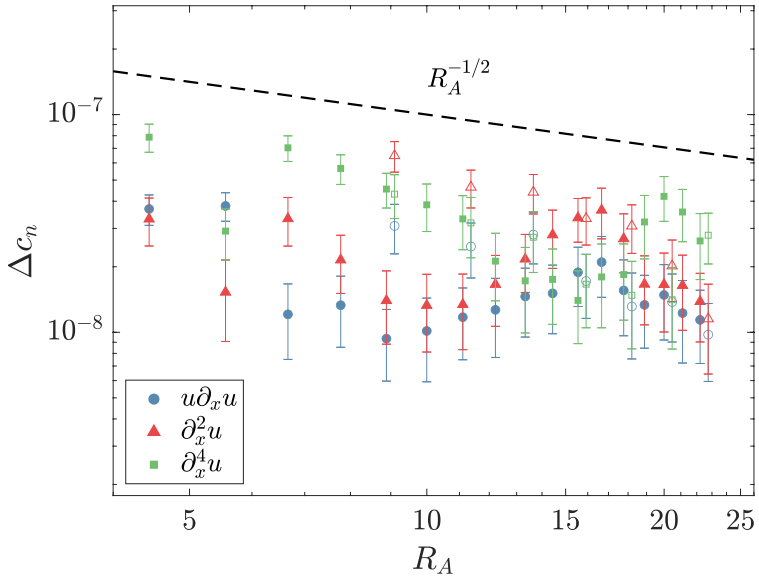


Figure 4.11: Parameter error as a function of area ratio, R_A. Filled markers correspond to fixing $F_t = 250$ and varying F_x; empty markers represent fixing $F_x = 32\pi$ and varying F_t. In this case $K = 500$, so as to ensure being toward the $K \gg R_A$ limit.

the second case, $L_x = 32\pi$ is set and L_t is varied. By changing the utilized domain of a pre-computed trajectory, the dependence on R_A is isolated from other factors dependent on the L_x used to compute the solution. After all, the domain size is the controlling parameter for the Kuramoto-Sivashsinsky system; the dynamics and time-averaged spectra could change as a function of L_x, which would corrupt the results of this numerical investigation. Broadly speaking, the predicted scaling is observed in Fig. 4.11, however there are significant deviations from $R_A^{-1/2}$ at the small R_A scale. This could be in part because though it is the same solution, just sampled differently, for every case, the information that results in a decrease in Δc_n may not be distributed evenly throughout the spatiotemporal domain. Understanding the small scale deviations from the broader predicted scaling would require further investigation which is outside the interest of this thesis.

4.4 Extension to Non-constant Parameters

Lastly, consider the case in which the physics changes throughout the spatiotemporal domain of the system. This non-uniformity of the governing equations could be due to non-

uniform boundary conditions, a time-varying forcing profile, or the boundary of the domain being non-parallel with external fields (e.g., a tilt in the Kolmogorov fluid containment causes the height of fluid to vary horizontally, which will make terms in the governing equation affected by vertical effects vary spatially as well). This breaking of translational invariance throughout the system's domain can be represented by the coefficients becoming functions of the independent variables, i.e.,

$$c_n \to c_n(\mathbf{x}, t).$$

The weak formulation approach can be modified to accommodate such scenarios, if the spatiotemporal variation can be represented by a linear combination of basis functions. On the contrary, particularly localized variations (eg, $\exp(-(x-x^*)^2)$, where x^* is the location of a "defect") are likely to be ill-approximated by basis functions if they are defined over a range of x large in comparison to width of the localized variation.

A term in a PDE with such coefficients takes the form

$$c_l(\mathbf{x}, t)\mathbf{F}_l,$$

where the parametric dependence has now been included. Then each coefficient can be written as

$$c_l(\mathbf{x}, t) \approx \sum_{m=0}^{M} c_m^{(l)} g_m(\mathbf{x}, t), \tag{4.37}$$

where $g_m(\mathbf{x}, t)$ is some basis function, and M should be chosen based on the how complicated the parametric dependence is, its flavor, and how well the dependence can be represented in the chosen basis (i.e., how quickly its spectrum $c_m^{(l)}$ decays with m). The library is now expanded for every l whose corresponding coefficient is suspected to have parametric dependence. To further illuminate the procedure, and to test its efficacy in the simplest scenarios, two examples are considered.

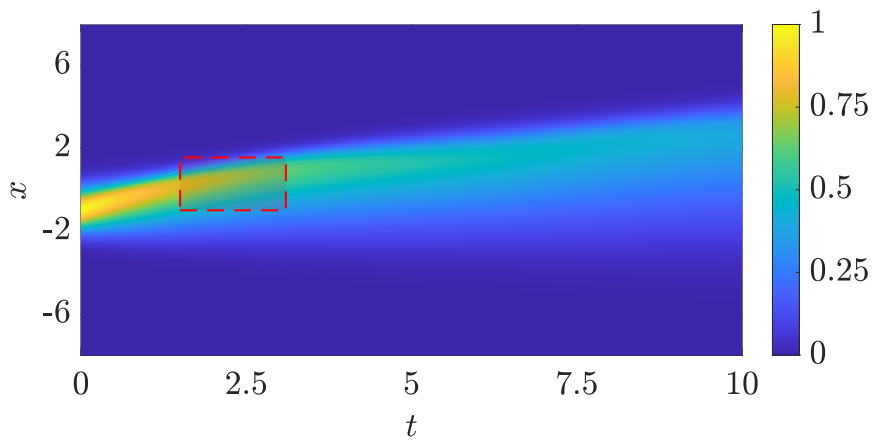


Figure 4.12: The trajectory used (without noise corruption) used to compute the temporal regularization. Overlaid in dashed red is the to-scale integration domain used to compute the library elements.

First, consider the Burgers' equation with diffusive regularization,

$$\begin{aligned} \partial_t u &= a(t) u \partial_x u + b \partial_x^2 u \\ a(t) &= -\left(1 + \frac{\sin(t)}{4}\right), \end{aligned} \tag{4.38}$$

with $b = 0.1$. The parametric dependence can be approximated using the Galerkin method above. Since the best basis functions may not be known *a priori*, the results for two different choices: Fourier basis and standard polynomials will be presented. For the system (4.38), the general model

$$\partial_t u = \sum_m c_m^{(1)} g_m(t) u \partial_x u + c_m^{(2)} g_m(t) \partial_x^2 u, \tag{4.39}$$

where

$$g_m(t) = \begin{cases} 1 & \text{if } m = 0 \\ \sin(mt) & \text{otherwise} \end{cases} \tag{4.40}$$

$$\text{or } = t^m \tag{4.41}$$

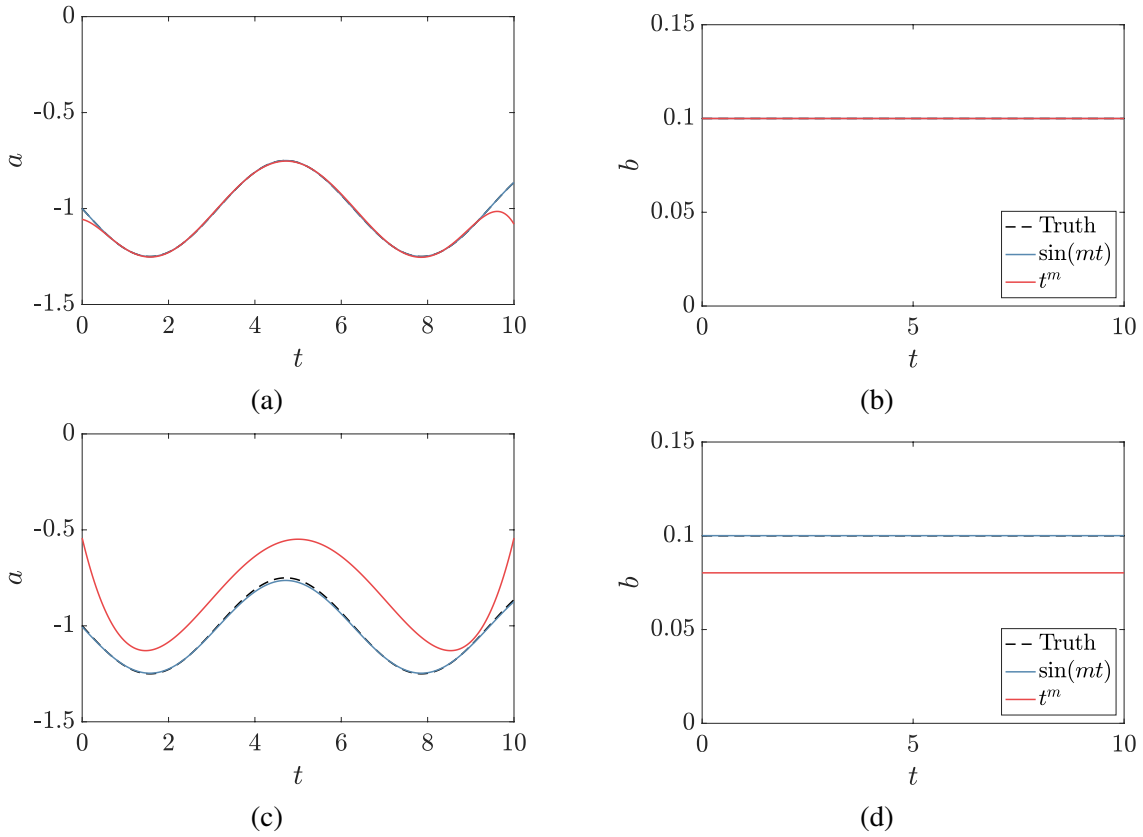


Figure 4.13: Top row: the parameters identified without noise corruption. Both fit the data remarkably well. Bottom row: identified parameters with 10% noise corruption. The good choice of basis function retains its strong fit, while the poor choice (polynomial) incurs large errors.

will be considered as a candidate, where t is shifted and scaled to be on $[-1, 1]$ for stability of fit for the polynomial case. For this example, and the one below, the (synthetic) systems have periodic boundary conditions, so a Fourier basis could have been guessed using domain knowledge; however, real systems are unlikely to have parametric variation of such neat forms.

The parametric dependence can complicate the transfer of derivatives away from the observed data, but they do not completely prevent it. For this example, the full set of library terms are given by

$$F_n = c_m^{(l)} g_m(t) \partial_x^{\nu_1} \left(u^{\nu_2}\right), \tag{4.42}$$

where $\nu_1 + \nu_2 \leq 4$. So the true terms correspond to $(\nu_1, \nu_2) = (1, 2)$ and $(2, 1)$. In this

case, the coefficients do not depend on the variable of which the derivatives are being taken with respect to, so evaluating the library terms can proceed as usual:

$$\begin{aligned} q_{kn} &= \langle w_j, F_n \rangle_i, \\ &= \int_{\Omega_i} w_j c_m^{(l)} g_m(t) \partial_x^{\nu_1}(u^{\nu_2}) d\Omega, \\ &= (-1)^{\nu_1} \int_{\Omega_i} c^{(l)} g_m(t) u^{\nu_2} \partial_x^{\nu_1} w_j d\Omega, \end{aligned} \tag{4.43}$$

where now n indexes unique pairs of l and m.

The two different sets of $g_m(t)$ are chosen to illustrate the difference between a good choice (sinusoides) and a bad choice (polynomial) of basis. In fact, the choice of sinusoides is extremely idealized, as it is the correct form of basis, but also windowed to integer multiples of the true regularization frequency, phase shifted correctly, and the parity is correct (ie, cosines are excluded). On the other hand, the choice of polynomials is ill-posed, as the forcing has too much variation over its duration than a polynomial can easily capture in a reasonable M. A trajectory was generated on the domain $x \in [-5, 5)$ and $t \in [0, 10]$, which was discretized into $N_x = 256$ grid points in space and $N_t = 256$ in time, using a spectral code; the generated trajectory is shonw in Fig. 4.12. Symbolic regression was applied to this trajectory with method parameters $2H_x/L_x = 2H_t/L_t \approx 0.16$, $\alpha = \beta = 4$, $I = 150$, and exponential weight function harmonics up to 2 in space and time (for weight function as in (4.15)). The estimated parametric dependence is shown in Fig. 4.13, compared with the truth.

For the noiseless case, the ideal basis reconstructs the parametric dependence indistinguishably from the truth. In fact, it identifies the model with residual

$$\eta = \frac{\|\hat{Q}\mathbf{c} - \mathbf{q}_0\|_2}{\|\mathbf{q}_0\|_2} \tag{4.44}$$

on the order of 10^{-6}, which is in line with what was observed for constant parameters in the

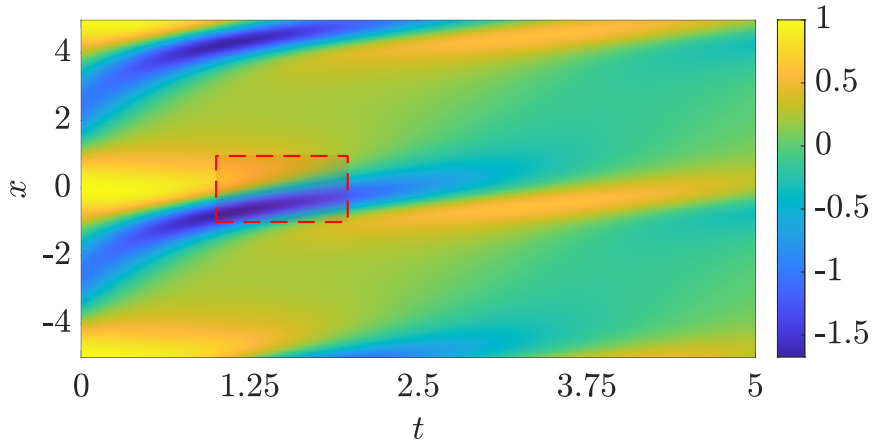


Figure 4.14: The trajectory used (without noise corruption) used to compute the spatial variation in parameters of the advection diffusion system. Overlaid in dashed red is the to-scale integration domain used to compute the library elements.

Kuramoto-Sivashinsky system considered earlier in the chapter. On the other hand, direct parameter comparison cannot be done for the polynomial basis. The polynomial used to fit the advection's time dependence was kept up to sixth order, and although it fits well in the middle, its edges diverge as could be expected of a poorly chosen basis. On the other hand, if the observed data is corrupted with Gaussian noise at each observation location with 0 mean and standard deviation equal to $0.1\langle|u(x,t)|\rangle_{x,t}$. This level of noise corruption would pose a serious problem for local approaches, due to the presence of the second order spatial derivative. However, given a good enough choice of basis function, the weak formulation can still recreate the temporal dependence extremely well. With a poor choice of basis, the corruption proves to be fatal, as shown in the bottom row of Fig. 4.13. Though the general shapes are correct, they exhibit significant offsets from the truth. Worse yet, for different realizations of integration domain ensembles, the results can be completely useless.

For systems with coefficients dependent on the same variable that the derivatives are with respect to, evaluating the library is slightly more complicated (moreso in concise implementation than anything else). To illustrate this, consider the spatially-dependent

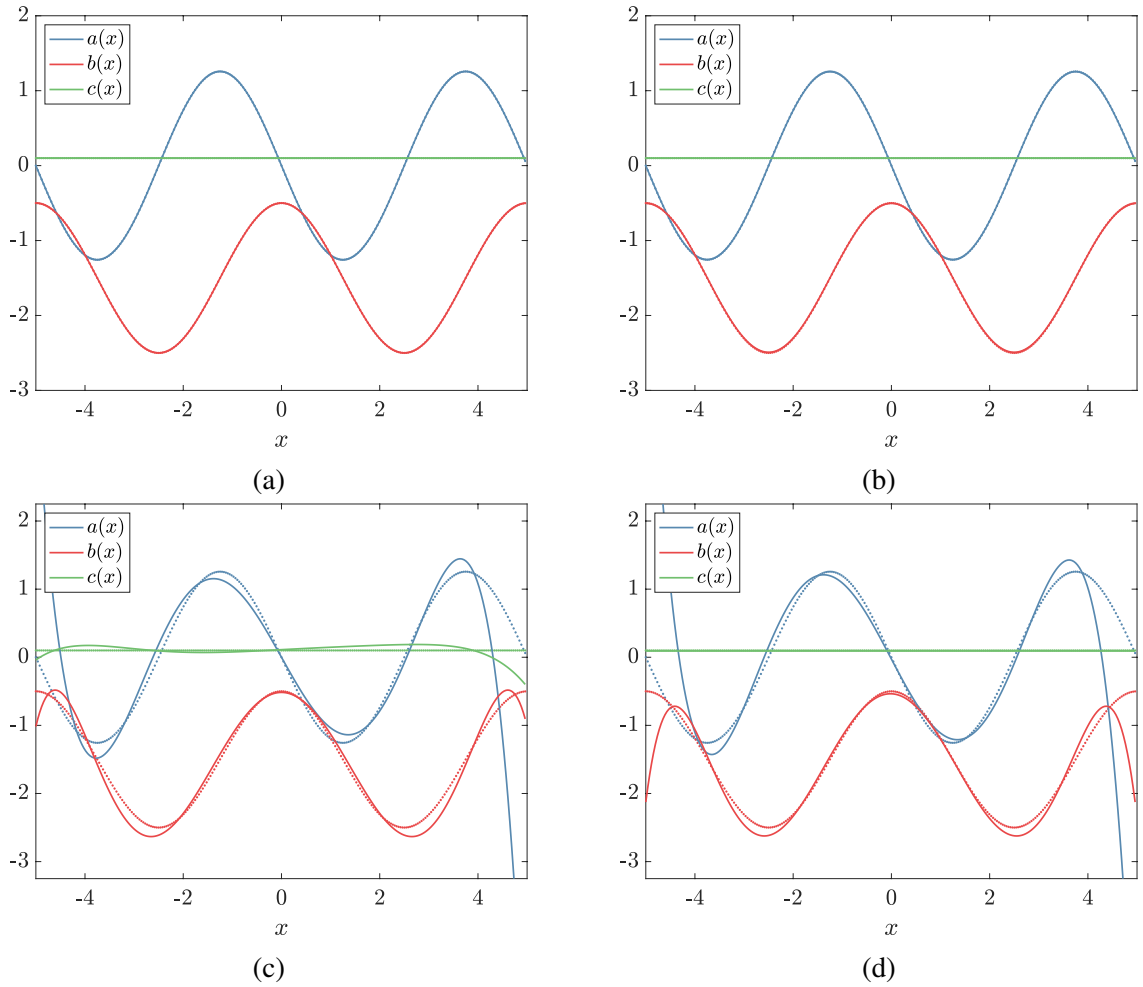


Figure 4.15: For both rows on the left a clean trajectory was used, and on the right the same trajectory corrupted with 10% noise was used. Top row: parameters identified in a Fourier basis. Bottom row: parameters identified in a polynomial basis. In all figures the solid lines represents the Galerkin fit, while the dots represent the true variation of the parameters.

advection-diffusion

$$\begin{aligned} \partial_t u &= \partial_x(v(x)u) + c(x)\partial_x^2 u, \\ &= a(x)u + b(x)\partial_x u + c(x)\partial_x^2 u, \end{aligned} \tag{4.45}$$

which represents a system that experiences spatially dependent diffusion. A trajectory from such a system was generated according to $c(x) = 0.1(1 + 1/2\cos(3\pi x/L_x))$, of the variable $u(x,t)$ subject to spatially dependent transport from $v(x) = -1.5+\cos(2\pi x/L_x))$,

for periodic spatial boundary conditions for $x \in [-5, 5)$ over a time period $t \in [0, 5]$. The trajectory's domain was discretized into $N_t = 256$ points in time and $N_x = 256$ in space.

The latter representation in (4.45) shall form the basis of the general candidate model (ie, only consider spatial dependence for the terms u, $\partial_x u$, and $\partial_x^2 u$), and the library terms will have their derivatives transferred as

$$\int_{\Omega_i} w_j c_3(x) \partial_x^2 u d\Omega \approx \int_{\Omega_i} w_j \sum_m c_m^3 g_m(x) \partial_x^2 u d\Omega = \int_{\Omega_i} u \partial_x^2 \left(w_j \sum_m c_m^{(3)} g_m(x) \right) d\Omega. \tag{4.46}$$

The results of the symbolic regression with method parameters $H_x \approx 0.1 L_x$, $H_t \approx 0.1 L_t$, $\alpha = \beta = 4$, $I = 150$, and complex exponential weight mode combinations up to first order in space and time are presented in Fig. 4.15 for both a clean trajectory and one corrupted with $0.1 \langle |u(x,t)| \rangle_{x,t}$ standard deviation Gaussian noise.

The Fourier basis again yield relative parameter error $\mathcal{O}(10^{-6})$, since its basis functions fit the true variation exactly, for the noise-less case and for the noise corrupted trajectoy yields a residual $\eta \sim \mathcal{O}(10^{-2})$. On the other hand, the polynomial basis estimations with $\eta \sim \mathcal{O}(10^{-2})$ for both noise-less and noisy cases. In this case the fit coefficients $c_m^{(l)}$ of higher order are fitting real dynamics and are less susceptible to noise corruption. Though, for fit order $M = 6$, the fit isn't very good on the edges, which is to be expected. Through these two examples, a proof-of-principle for spatial/temporal-dependent model identification has been presented. A brief investigation into the largest anticipated complication for this type of approach, choosing the right basis, was also given. Further developing this approach could prove important/necessary for identifying systems that are affected by spatiotemporal control mechanisms, which is an ultimate goal of the Kolmogorov experimental set-up.

In this chapter, an alternative to constructing the library matrix by local evaluations of the constituent terms in the governing equation to be discovered has been presented. Based on the results above, this alternative promises to enable experimental application of existing

symbolic regression techniques via its remarkable robustness to noise pollution and grid density. It was also shown to be able to handle special cases of non-constant coefficients. However, some outstanding issues still remain. The approach needs to be extended to the higher dimensional systems, and it needs to be able to handle the latent variables like the direct approach was able to in Chapter 3.

CHAPTER 5

WEAK FORMULATION: 2D GENERALIZATION AND LATENT VARIABLES

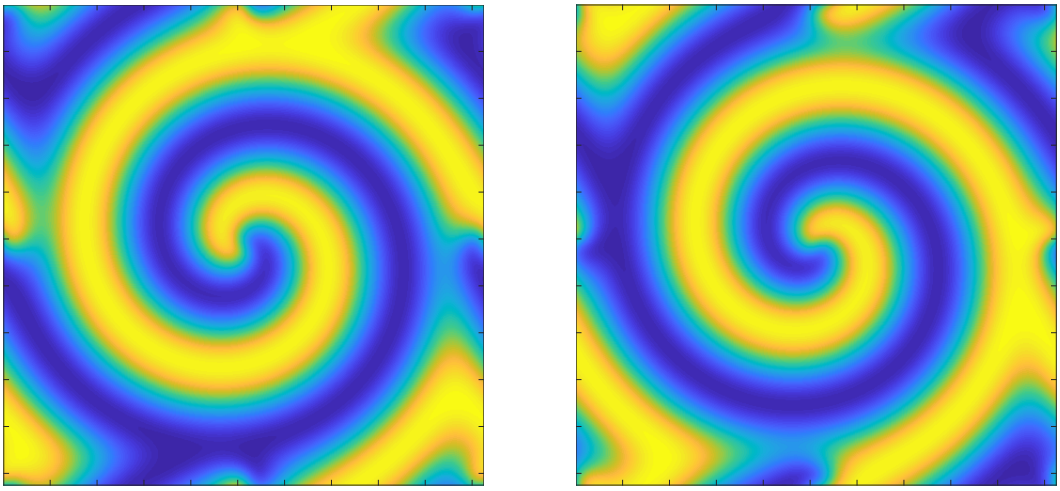
The approach has thus far been limited to a simple test case to illustrate its conceptual details, and discuss its dependencies on method parameters and data fidelity. How the method scales with the grid density, external noise corruption, number of integration domains, and overall information utilized was shown. In the prior two cases the method is head and shoulders above alternative local evaluations such as those developed in Chapter 3. The present chapter will show how systems with larger libraries and multiple state variables can be dealt with, and how well the method handles such systems.

Afterward, the quasi-2D fluid flow system shall be returned to, and the existing methods shall be taken to real data. For now, the library crafted from physics and symmetry constraints in Chapter 2 shall be inherited, but a special weight function needs be designed to address the problems associated with symbolic regression of the Kolmogorov-like flow. After general results are presented, some of the scaling laws shall be recreated and validated across systems. By the end of the chapter, everything necessary to tackle experimental data shall be available.

5.1 Reaction Diffusion

To demonstrate the method's applicability to sparse regression problems, consider the $\lambda-\omega$ reaction-diffusion system [108] in two spatial dimensions,

$$\begin{aligned} \partial_t u &= D\nabla^2 u + \lambda u - \omega v, \\ \partial_t v &= D\nabla^2 v + \omega u + \lambda v, \end{aligned} \tag{5.1}$$



(a) (b)

Figure 5.1: A typical snapshot of the fields (a) u and (b) v for the λ-ω reaction diffusion system. The x axis is horizontal and the y axis is vertical.

where $\omega = -\beta(u^2 + v^2)$, $\lambda = 1 - u^2 - v^2$, and $\beta = 1$ and $D = 0.1$ are constants. This system can be cast in the form of Eq. (1.1) by defining a vector $\mathbf{u} = [u, v]$. To test the weak formulation approach, sparse regression was applied to a generalization of (5.1), where the reaction terms are given by polynomials in u and v up to third order. In total, the generalized model involves a total of 20 different terms (two diffusion terms and 18 polynomial terms). Correspondingly, 20 unknown coefficients need to be determined.

The sparse regression problem for the $\lambda - \omega$ system can be block-diagonalized by using a weight function $\mathbf{w} = [w, 0]$ (or $\mathbf{w} = [0, w]$) to reconstruct the first (or second) equation in (5.1), yielding two independent linear systems (1.4) with 10 library terms each. The integration domains Ω_k are three-dimensional. The integrals involving terms such as $u^\alpha v^\beta$ do not require integration by parts, while the two integrals involving the Laplacian terms are integrated by parts twice to get rid of derivatives on u and v, e.g.,

$$q_{k1} = \int_{\Omega_k} w \nabla^2 u \, d\Omega = \int_{\Omega_k} u \nabla^2 w \, d\Omega. \tag{5.2}$$

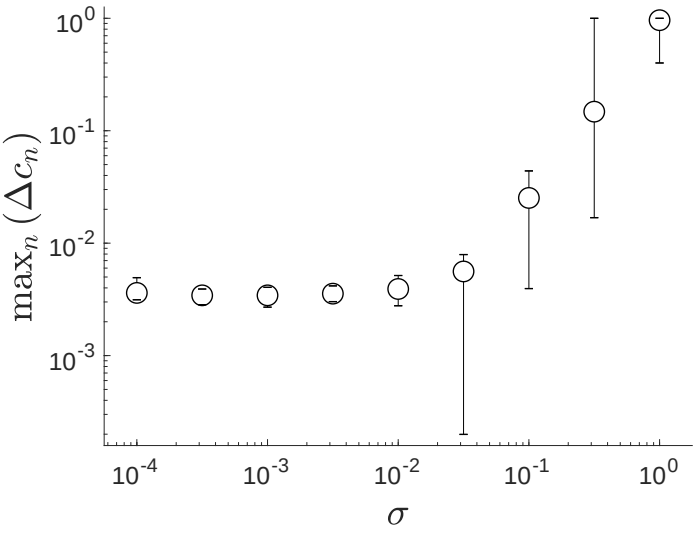


Figure 5.2: The accuracy of parameter reconstruction for the $\lambda - \omega$ reaction-diffusion system as a function of the noise amplitude. Shown is the largest error, which corresponds to one of the diffusion coefficients.

In both cases, the corresponding boundary terms vanish if we choose

$$w = E_\alpha(\underline{x})E_\beta(\underline{y})E_\gamma(\underline{t}), \tag{5.3}$$

where $\alpha, \beta \geq 2$ and $\gamma \geq 1$; here is chosen $\alpha = \beta = 2$ and $\gamma = 1$. These exponent values are relatively small; however, the grid density for this synthetic system (as will be detailed shortly), is relatively high.

The surrogate data was obtained by computing the solution of (5.1) using the integrator employed in Ref. [20]; a typical snapshot is shown in Figure 5.1. The computational domain of size $L_x = 20$, $L_y = 20$, $L_t = 10$ was discretized using a grid with spacing $\Delta x_c = \Delta y_c = 0.0391$ and $\Delta t_c = 0.05$, and Gaussian random noise with standard deviation σ was added to both u and v at each grid point. The data was input to the weak formulation procedure at full resolution; as mentioned, it is well-resolved at approximately 170 grid points per spatial wavelength of the spiral solution. The dimensions of the integration domains Ω_k were chosen as $H_x = H_y \approx 1$ and $H_t = 1.25$. The results of sparse regression are shown in Figure 5.2. The results are again quantified by Δc_n (3.14), but presenting

the noise dependence of the parameter error for each library term is cumbersome for a single plot. Instead, the worst parameter error estimate is shown. This corresponds to the diffusion library terms ($\nabla^2 u$, and $\nabla^2 v$), who have the smallest R_n value, which further illustrates the discussion in Section 3.3. The full parameter estimates are given in Table 5.1 for a particular noise amplitude of 1%.

For the same data sets for which Figure 5.2 was crafted from, it is found that, for noise levels of up to 5%, the model was reconstructed correctly (with no spurious or missing terms) for each distribution of Ω_k in the ensemble, with all parameters estimated to an accuracy of better than 1%. With 10% noise, the model is identified correctly in about 95% of cases, and at 30% noise, the model is identified correctly in about 20% of cases, with the remaining cases featuring spurious terms (linear in u and v) that are not present in the $\lambda - \omega$ model. For reference, sparse regression based on local evaluation of derivatives [20] failed to correctly identify this model, generating spurious terms in the presence of as little as 1% noise. It should be noted that using ensemble sparse regression makes it easy to detect the presence of spurious (missing) terms and eliminate (add) them while still preserving the accuracy with which all of the correct terms are estimated (in this case, about 3% for the worst case offenders with 10% noise). It is also worth pointing out that, unlike the standard approach [20], weak formulation requires no intermediate noise reduction.

Table 5.1: List of coefficients determined for the $\lambda - \omega$ reaction-diffusion system.

Term	u (original)	u (estimated)	v (original)	v estimated)
∇^2	0.1	0.0995	0.1	0.0995
u	1	0.998	0	0
v	0	0	1	0.9989
u^2	0	0	0	0
uv	0	0	0	0
v^2	0	0	0	0
u^3	-1	-0.9988	-1	-1.0004
u^2v	1	0.9996	-1	-0.9984
uv^2	-1	-0.9986	-1	-1.0003
v^3	1	1.0006	-1	-1.0003

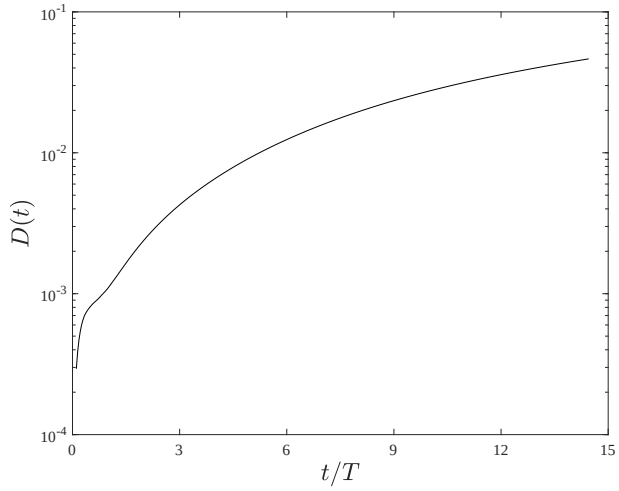
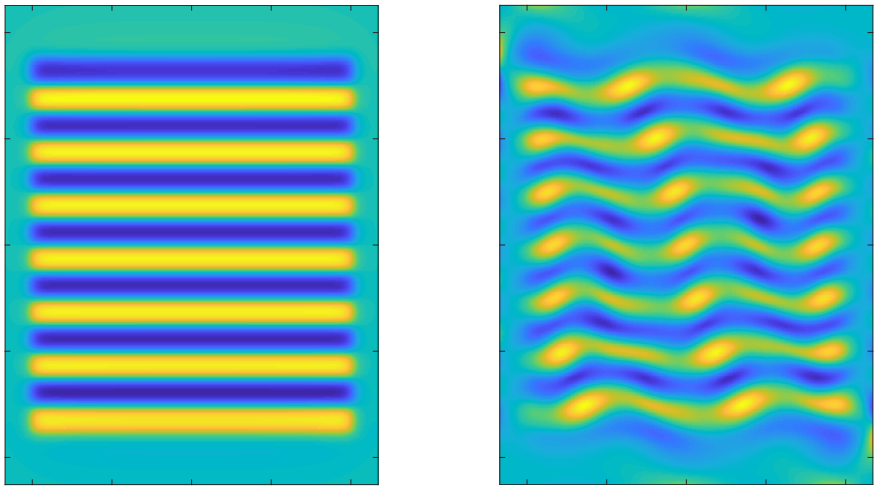


Figure 5.3: Prediction error for the λ-ω reaction-diffusion system.

To test how well the identified model predicts the evolution of the system, rather than exclusively measuring the accuracy of the model in terms of parameter error, again consider the normalized distance $D(t)$ from (4.17). For the test, the model was reconstructed from synthetic data with 1% noise, which yielded the coefficients, shown in Table 5.1, with a precision of 0.5% or better and no spurious or missing terms. The prediction error for this model is shown in Figure 5.3. This system is not chaotic, so the error does not grow exponentially quickly and the Lyapunov time is undefined. A proper time scale is rather one period of rotation of the spiral wave solution, $T = 6.94$. We find that the model yields accurate predictions (i.e., $D(t) < 0.01$) over a temporal interval of length $5.23T$, also in line with what can be expected for a near-perfect model.

5.2 Kolmogorov Flow

The method has been illustrated using a one-dimensional dynamical system, and it is also proven to work together with sparse regression on a two-dimensional system with multiple state variables along with a large library. Now return to the quasi-2D flow example to demonstrate how a weight function can interact with a vector PDE, and how the weight function can be chosen specially to remove latent variables.



(a) (b)

Figure 5.4: The Kolmogorov-like flow: (a) the forcing profile f and (b) snapshot of the vorticity $\omega = \partial_x u_y - \partial_y u_x$. The x axis is horizontal and the y axis is vertical.

To convert (2.14) to weak form, multiply it by a vector field $\mathbf{w}$ and integrate the result by parts over the three-dimensional spatiotemporal domain Ω_k of size $2H_x \times 2H_y \times 2H_t$. Assuming again that the boundary terms vanish, for the linear terms:

$$\begin{aligned} q_{k0} &= \int_{\Omega_k} \mathbf{w} \cdot \partial_t \mathbf{u} \, d\Omega = -\int_{\Omega_k} \mathbf{u} \cdot \partial_t \mathbf{w} \, d\Omega, \\ q_{k2} &= \int_{\Omega_k} \mathbf{w} \cdot \nabla^2 \mathbf{u} \, d\Omega = \int_{\Omega_k} \mathbf{u} \cdot \nabla^2 \mathbf{w} \, d\Omega, \\ q_{k3} &= \int_{\Omega_k} \mathbf{w} \cdot \mathbf{u} \, d\Omega. \end{aligned} \tag{5.4}$$

The nonlinear term can be rewritten in a similar way using the incompressibility condition $\partial_i u_i = 0$ (where summation over repeated indices is implied):

$$\begin{aligned} q_{k1} &= \int_{\Omega_k} w_i u_j \partial_j u_i \, d\Omega = -\int_{\Omega_k} u_i \partial_j (w_i u_j) \, d\Omega \\ &= -\int_{\Omega_k} u_i u_j \partial_j w_i \, d\Omega = -\int_{\Omega_k} \mathbf{u} \cdot (\mathbf{u} \cdot \nabla) \mathbf{w} \, d\Omega. \end{aligned} \tag{5.5}$$

Finally, for the terms involving the latent variables:

$$q_{k4} = \int_{\Omega_k} \mathbf{w} \cdot \nabla p \, d\Omega = -\int_{\Omega_k} p \nabla \cdot \mathbf{w} \, d\Omega,$$
$$q_{k5} = \int_{\Omega_k} \mathbf{w} \cdot \mathbf{f} \, d\Omega. \tag{5.6}$$

In order for the boundary terms to vanish on a rectangular domain Ω_k centered at (x_k, y_k, t_k), it is necessary to have $\mathbf{w} = 0$ on $\partial\Omega$, as well as $\partial_x \mathbf{w} = 0$ at $\underline{x} = \pm 1$ and $\partial_y \mathbf{w} = 0$ at $\underline{y} = \pm 1$, where the underbar denotes rescaled variables $\underline{x} = (x - x_k)/H_x$, $\underline{y} = (y - y_k)/H_y$, and $\underline{t} = (t - t_k)/H_t$. Next, the dependence on the pressure field and the steady forcing can be eliminated by additionally requiring that

$$\nabla \cdot \mathbf{w} = 0 \tag{5.7}$$

and

$$\int_{-1}^{1} \mathbf{w} \, d\underline{t} = 0. \tag{5.8}$$

All of the above conditions on $\mathbf{w}$ can be satisfied by setting $\mathbf{w} = \nabla \times (\psi \hat{z}) = \hat{x}\partial_y \psi - \hat{y}\partial_x \psi$, where

$$\psi = \sin(\pi \underline{t})(\underline{x}^2 - 1)^{\alpha}(\underline{y}^2 - 1)^{\beta} \tag{5.9}$$

and $\alpha = \beta \geq 3$ ($\alpha = 3$ was used in this study). This yields $q_4^k = q_5^k = 0$ and

$$
\begin{aligned}
q_{k0} &= -\int_{\Omega_k} (u_x \partial_y - u_y \partial_x) \partial_t \psi \, d\Omega, \\
q_{k1} &= \int_{\Omega_k} \left[u_x u_y (\partial_y^2 - \partial_x^2) + (u_x^2 - u_y^2) \partial_{xy} \right] \psi \, d\Omega, \\
q_{k2} &= \int_{\Omega_k} (u_x \partial_y - u_y \partial_x) \nabla^2 \psi \, d\Omega, \\
q_{k3} &= \int_{\Omega_k} (u_x \partial_y - u_y \partial_x) \psi \, d\Omega.
\end{aligned} \tag{5.10}
$$

The linear system (1.4) can now be constructed just like in Chapter 3, but this time by evaluating the integrals in (5.10) over a set of domains Ω_k. Note that this linear system involves neither the derivatives of the noisy observable data (components of the $\mathbf{u}$ field) nor the latent variables (p and $\mathbf{f}$ fields).

To test the approach, surrogate data for $\mathbf{u}$ was generated again by solving the incompressible version of (2.14) with the (non-dimensional) parameters $c_1 = -0.826$, $c_2 = 0.0487$, $c_3 = -0.157$, and $c_4 = 0$, which correspond to the experimental setup described in Ref. [92]. The forcing generates a weakly turbulent flow (a representative snapshot is shown in Figure 5.4(b) as a reminder), which was computed using the numerical integrator described in Ref. [92] on a domain of size $L_x = 14$, $L_y = 18$, $L_t \approx 920$ and a computational grid with $\Delta x_c = \Delta y_c = 0.025$ and $\Delta t_c \approx 0.02$. The data was then subsampled on a coarser grid with spacing $\Delta x = \Delta y = 0.1$ and $\Delta t = 0.2302$, and mean-zero Gaussian random noise with variance σ was added to both components of the flow velocity $\mathbf{u}$. The integrals in (5.10) were evaluated over domains Ω_k of size $H_x = 11.2$, $H_y = 14.4$, and $H_t \approx 34.5$, centered around randomly chosen points $\mathbf{x}_k$ shown in figure 5.5. Though this uses a lot more extended information per library element q_{kn}, the resolution of the grid is far inferior to that which was necessary for the direct approach to be operable.

As Figure 5.6 illustrates, the approach successfully reconstructs the parameters for the incompressible version of (2.14). Just like in the case of the Kuramoto-Sivashinsky equa-

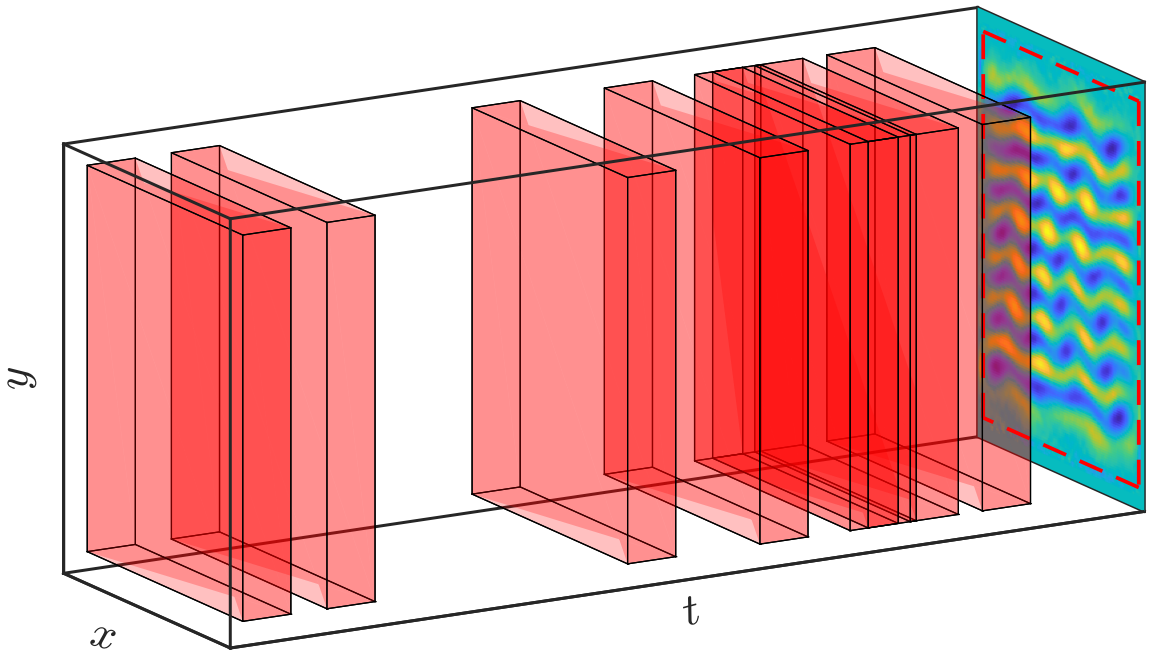


Figure 5.5: Sampling scheme for the integral approach for the Kolmogorov system. The spatial sizes of the integration domains are to scale, while again the temporal length of the turbulent trajectory is too long to visualize to scale. Note that the domains are so large in the spatial directions that they only vary in the temporal direction. Additionally, notice that overlap is not prohibited due to the temporal extent of the data with respect to the number of samples K. When comparing to the size of the local fitting neighborhoods in figure 3.1, note that although the physical size of the domains are much smaller, the required density of data points is much higher in that case, as the grid points shown here are only sub-sampled by a factor of 10^2. In the back is an example snapshot at the relevant Re.

tion, noise up to 10% does not meaningfully affect the accuracy of model reconstruction, with the coefficients c_1, c_2, and c_3 estimated to within 1% or better. In fact, even with 100% noise, the coefficients can still be estimated to within roughly 10%. For reference, experimental data [92] obtained using particle image velocimetry has roughly 3% noise, at which level local sparse regression failed completely (see Figure 3.8). One more, to supplement the parameter error, the time it takes for trajectories with disparate parameter sets to separate (in terms of D) when starting from the same initial conditions is compared with the Lyapunov time. For this system, the Lyapunov time is $\tau = 23.02$, and the prediction error is shown in Figure 5.7. It is found that the model yields accurate predictions (i.e., $D(t) < 0.01$) over a temporal interval of length 0.88τ, which is again in line with what can

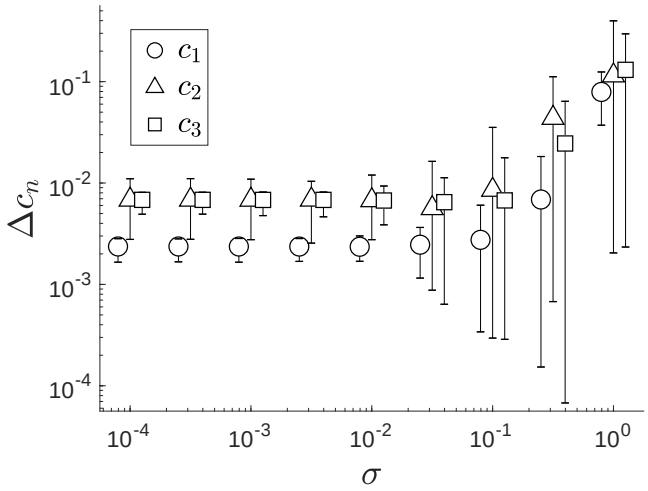


Figure 5.6: The accuracy of parameter reconstruction for the 2D Kolmogorov-like flow model as a function of the noise amplitude.

be expected for a perfect model due to the chaotic nature of the dynamics.

Now that the method has been devised within the context of the Kolmogorov flow, and its efficacy has been demonstrated, some of the scaling laws determined previously can be discussed. In particular, the dependence on the grid spacing is presented. Firstly, the trajectory used is generated for the incompressible model with (dimensional) parameters $c_1 = 0.826$, $c_2 = 3.2312 \times 10^{-6}$ m^2/s, and $c_3 = 0.0645$ Hz. The grid spacing used for integration was $\Delta x_c = \Delta y_c = 1/40$ and $\Delta t_c = 1/64$ s. The trajectory was initialized from random noise, and let evolve for 1000 s, at a current value of $I = 20.01$ mA. The velocity fields u, v were linearly interpolated in the y and x direction, respectively, so that they were on the same grid, as they are needed to be for the direct evaluation of the advection term's integral.

The grid spacing in each direction shall be varied together, and each shall be down-sampled such that in non-dimensional units they are all approximately equal to the same non-dimensional grid spacing value h. The time-scale chosen to non-dimensionalize Δt was the correlation time, which is approximately 20 s. Furthermore, the method parameters for the weak formulation are as follows: the integration domain lengths were $2H_x = 3.5$, $2H_y = 3.5$, and $2H_t = 50$s; the envelope exponents were $\alpha = \beta = \gamma = 6$; the weight func-

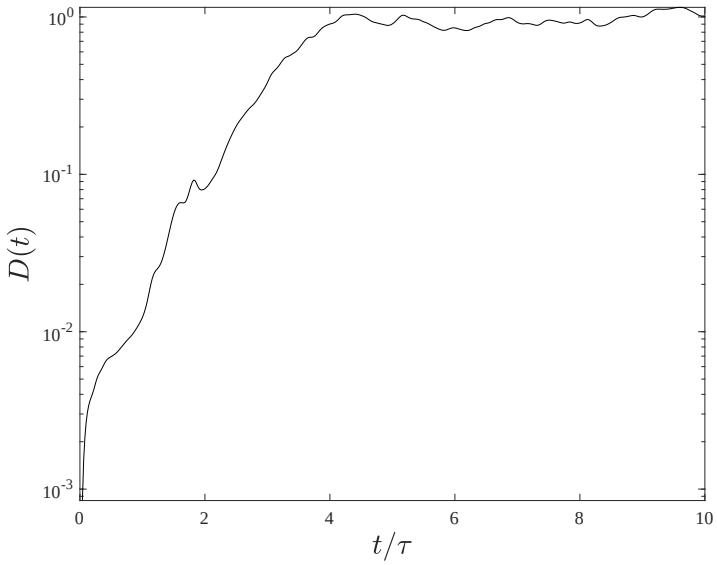


Figure 5.7: Prediction error for the Kolmogorov flow.

tion was modified with Legendre polynomials, so that more information could be sampled for the same integration domain, as in Section 4.3 (this weight function will be discussed in further detail in the next chapter); for each trial 100 integration domains are used, and the process is repeated for 30 trials for each value of h. The results of the numerical experiment are shown in Figure 5.8. Since the interest here is on the scaling of the method with the grid spacing of the available data, the model is assumed to be known, and the only interest is the error introduced in the estimated parameters as a function of sampling sparsity.

The previous arguments based on the error in numerical quadratures can be extended here. As before, the error contained in a given term scales with how many derivatives the integrand has equal boundary terms. In this case, inspecting the forms of the library terms,

$$q_{k1} = \int_{\Omega_i} [uv(\partial_y^2 - \partial_x^2) + (u^2 + v^2)\partial_{xy}]\psi_j d\Omega \tag{5.11}$$

$$q_{k2} = \int_{\Omega_i} (u\partial_y - v\partial_x)\nabla^2\psi_j d\Omega \tag{5.12}$$

$$q_{k3} = \int_{\Omega_i} (u\partial_y - v\partial_x)\psi_j d\Omega \tag{5.13}$$

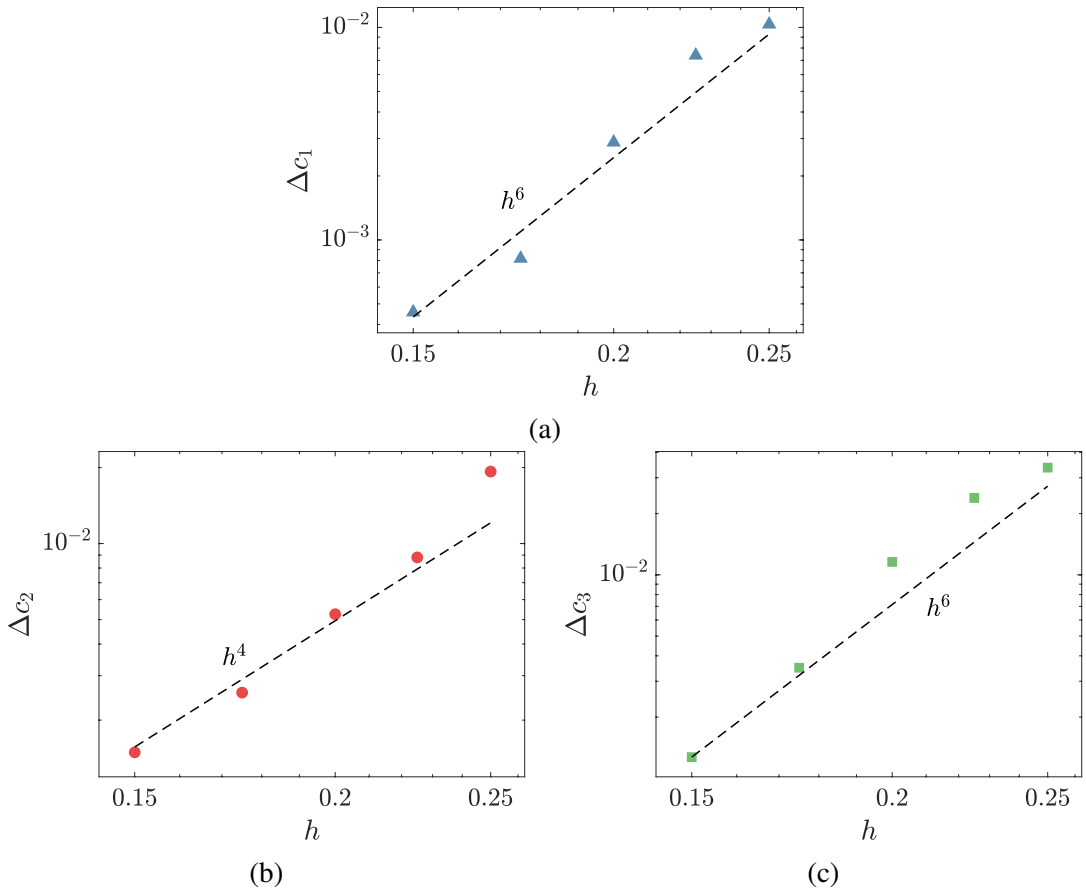


Figure 5.8: The scaling derived for the Kuramoto-Sivashinsky example is observed again for the Kolmogorov flow. The data were generated for $h = \Delta x = \Delta y = 1/40 \approx \Delta t$, and all three grid spacings downsampled to the values shown.

it can be seen that for these three terms the exponent factor $\mu = \min(\alpha - \nu_x, \beta - \nu_y, \gamma - \nu_t)$ varies between, 4, 3, and 5, respectively. Hence the parameter errors should scale as

$$\Delta c_1 \sim h^6, \qquad \Delta c_2 \sim h^4, \qquad \Delta c_3 \sim h^6. \tag{5.14}$$

Indeed this scaling is observed in Figure 5.8, for the range of h considered. Notice the discrepancy between the smallest h in the figures ($h = 0.15$) and the spacing at which the data was generated ($h = 0.025$). For $0.025 \leq h \leq 0.15$ the parameter error Δc_n is static, as opposed to the expectation that the accuracy would improve with a finer mesh. One potential explanation is that the rudimentary interpolation of the state variables onto

a collocated grid (from the staggered grid on which they were generated) introduces a ceiling for the accuracy with which they can solve the governing equations (and thus the weak version). Another source of error could be due to a limit on the accuracy with which the integration scheme produces data that satisfies the weak formulation of their governing equations. Since the experimental data is acquired already on a collocated grid, and errors in the PIV estimates of the horizontal velocities are wholly unrelated to errors introduced by a particular choice of integration scheme, this problem was not investigated further. This error floor also prevents a more thorough investigation into the noise dependence than what is presented in Figure 5.6, though the predicted linear dependence can still be observed for $\sigma \geq 10^{-1.5}$.

CHAPTER 6
EXPERIMENTAL APPLICATION

The efforts presented thus far in this thesis culminate in this chapter. Herein the weak formulation method of symbolic regression is applied to experimentally acquired Kolmogorov flow data. This is a non-trivial instance of true system-discovery, whereas other methods have only seen success on synthetic data. Before being able to perform the regression process, it must be acknowledged that certain library terms in the general model (2.14) are not completely compatible with the derivative transferral procedure. Some derivatives will remain on the observed data. As such, they must be evaluated numerically. The first section details that procedure, and by what methods it is optimized. Afterward, the model estimates for the Kolmogorov system over a range of Re is presented. Finally, the chapter concludes with demonstrating that the destructive measures of the weight function constraints can be counteracted via latent field reconstruction.

6.1 Sparse Regression with Kolmogorov Flow

As discussed in section 2.3, the Kolmogorov flow model can be generalized to include the additional terms $(\nabla \cdot \mathbf{u})\mathbf{u}$, $(\nabla \cdot \mathbf{u})^2\mathbf{u}$, $(\nabla \times \mathbf{u})^2\mathbf{u}$, and u^3. The sum of which terms can be understood as additional effects due to the inherent three-dimensionality of the fluid, the intensity of which should increase with the forcing. The nonlinear terms that contain derivatives that cannot be completely eliminated demand a procedure for evaluating the required derivatives. In this case, the derivatives do not exceed first order in space, so the burdens discussed in detail in chapter 3 are alleviated to an extent. Because the spatial derivative information is available on a regularly gridded global domain, evaluating derivatives via a filtered Fourier method is an expedient choice.

The physical domain used in the integrations so far has been approximately 80% of the

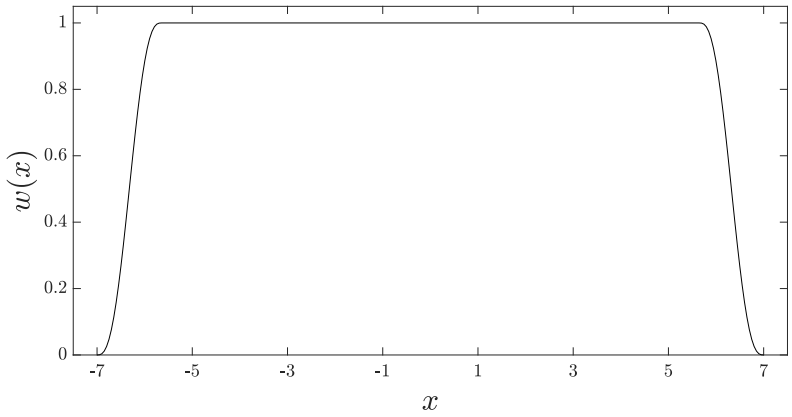


Figure 6.1: The windowing function in the x direction, where $\Omega_x = 14$.

full extent, in each spatial direction. A windowing function was multiplied to the velocity fields, where the windowing function is the prodcut of two one-dimensional functions defined by

$$w(x) = \begin{cases} p(x), & \text{if } 0 \leq x \leq 0.2L_x, \\ 1, & \text{if } 0.2\Omega_x \leq x \leq 0.8L_x, \\ p(-x - L_x), & \text{if } 0.8L_x \leq x \leq L_x, \end{cases} \tag{6.1}$$

and is shown in figure 6.1. The function $p(x)$ is some polynomial whose coefficients are chosen such that $\partial_x^m p(x)|_{x=0} = 0, \forall m \in \{0, 1, 2, 3\}$, $p(0.2L_x) = 1$, and $\partial_x^m p(x)|_{x=0.2L_x} = 0, \forall m \in \{1, 2, 3\}$. Afterward, a 2D spectra at each point in time is obtained for the windowed velocity field using a discrete Fourier transform. The spectra was then low-pass filtered by setting all wave-numbers with index above a certain integer, labeled cut, to 0.

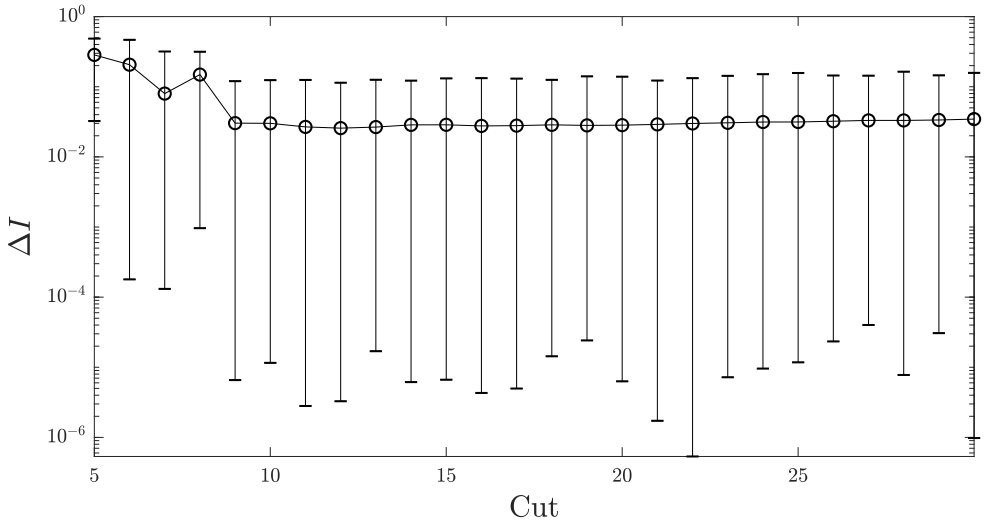
To identify the optimal value for cut, the following metric is considered

$$\Delta I = \frac{\langle |I_{FD}(t) - I_F(t; cut)| \rangle_t}{\langle |I_{FD}(t)| \rangle_t}, \tag{6.2}$$

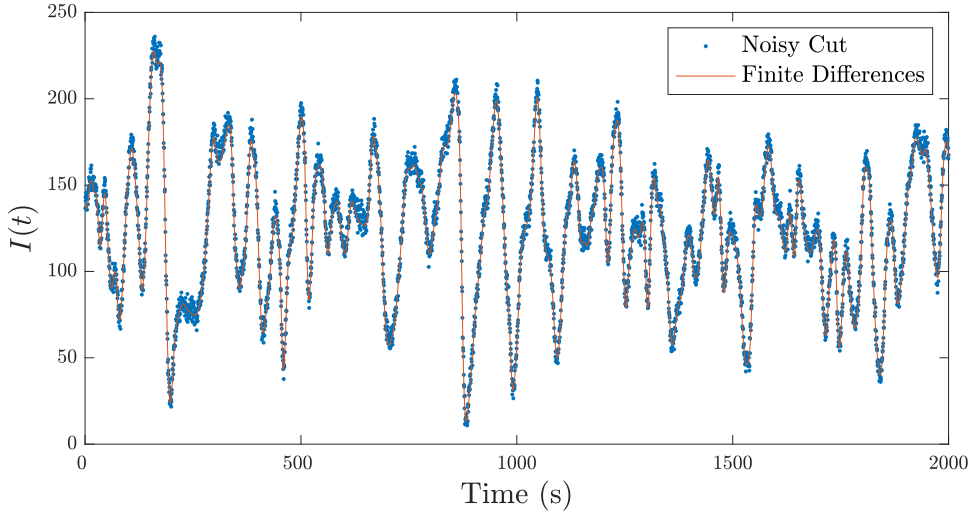
where

$$I_{...}(t) = \int_{-H_x}^{H_x} \int_{-H_y}^{H_y} \partial_y u(x, y, t) dy dx \tag{6.3}$$

is the integral of the first order derivative at a particular time t, and the subscript denotes



(a)



(b)

Figure 6.2: Shown in (a) is the value of ΔI as a function of the spectral window width, cut. Errorbars here represent the entire range of the data. Eventually ΔI will increase, as for large cut the wavelengths most critically corrupted by noise will be amplified unduly be the derivative operation. In (b) is the value of the integrals as a function of time. The orange solid line represents the "truth," as the derivatives in the integrand are calculated from noise-less data using a central differences scheme. Alternatively, the blue dots represent the integrals calculated with derivatives from noisy data using the Fourier procedure described in the text at $cut = 15$.

the method by which the derivative was computed: FD corresponds to finite differences applied to noise-less data, and F corresponds to a discrete Fourier method using the spectral cut applied to data corrupted by 5% additive Gaussian noise. Hence, the cut is chosen to

optimize the types of variables to be used in the method, rather than the local derivative values themselves. In figure 6.2a, the value for ΔI is shown as a function of the value for cut, while in Figure 6.2b the estimated integral values are compared with their noise-less counterparts over an extended period of time.

The value of $cut = 15$ is chosen, although based on Figure 6.2a any $cut \geq 9$ appears to be a reasonable choice. Also, cut was chosen for all spatial derivatives for both velocity fields u, v, based only on $\partial_y u$. This choice was made because the magnetic force acts in the $\hat{x}$ direction, appearing in the governing equation for u. Additionally, the forcing profile has periodicity in the y direction. So it was expected that the most spectral information would be carried in the y direction of the u field. Note that range of cut considered is too small to include the inevitable increase in ΔI at higher thresholds due to noise corruption.

6.2 Application to PIV Data

Using particle image velocimetry (PIV), values for the horizontal velocity at the height of the dielectric-electrolyte fluid interface are obtained for a range of forcing strengths, quantified by their effect on the induced flow, which is encoded by the dimension-less parameter the Reynolds number:

$$Re = \frac{UL}{\bar{\nu}} = \frac{\langle \sqrt{\langle u^2 + v^2 \rangle_{x,y}} \rangle_t w}{\bar{\nu}}, \tag{6.4}$$

where U is a velocity scale chosen as the long-time average of the root-mean-square velocity, $L = w$ is a length scale chosen as the width of one of the bar magnets, and ν is a viscosity obtained by the depth-averaging procedure presented in (2.6). The spatial snapshots of the horizontal velocity are obtained every second, $\Delta t = 1$ s, on a regular spatial grid, with Δx for the different data sets shown in Table 6.1. Typical flow profiles for u and v are shown in Figure 6.3. The noise present in the data has been experimentally characterized from the deviations from the long-time averaged low Re base state. The noise

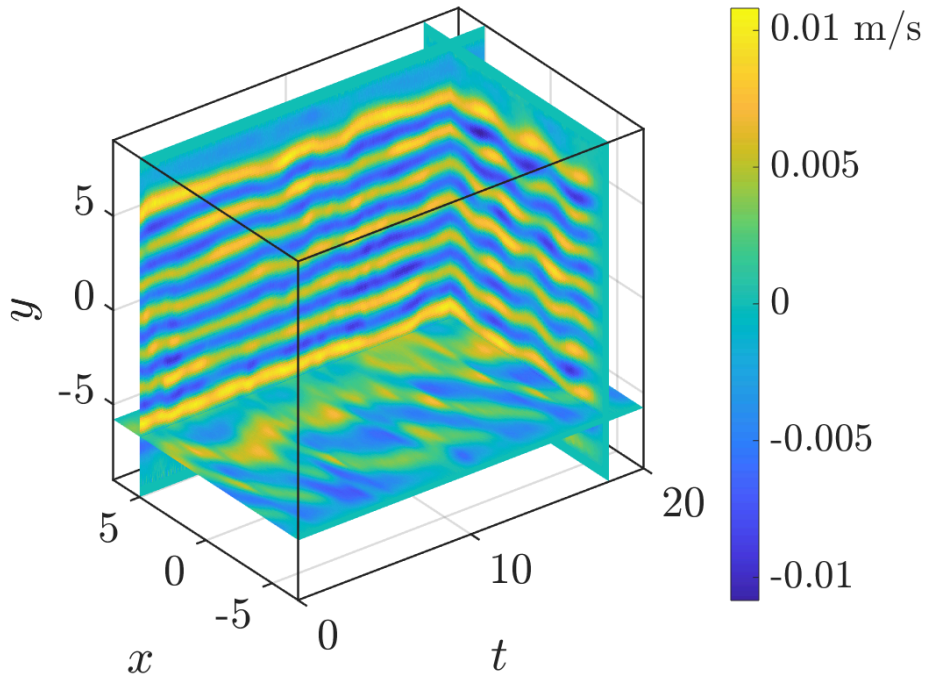
has zero mean and standard deviation $\approx 0.05u_0$, where u_0 is the velocity magnitude. Other relevant features of the data, such as dynamical time-scales, and spatial and temporal extent of the data, are given in Table 6.1. As can be seen, the data are analyzed over a variety of circumstnaces; this should test, in part, how robust the weak formulation method is to the the conditions of the data.

The resolutions and noise level are well within the regime for which the weak formulation yields accurate system identification for synthetic data; hence, if the library is sufficiently large, a model for the experimental system should be identified with reasonable accuracy. However, some of the data sets are fairly short $\mathcal{O}(10)\tau$, where τ is the period if the dynamics are periodic and the correlation time if the dynamics are chaotic, so there are few unique integration domain choices, since $2H_t \approx 4\tau$. To counteract this, in addition to sampling the data with unique choices of integration domains, the weight functions are varied several times for each integration domain. The Legendre polynomials were emperically found to be a good choice, which modifies the weight's base functional part to be

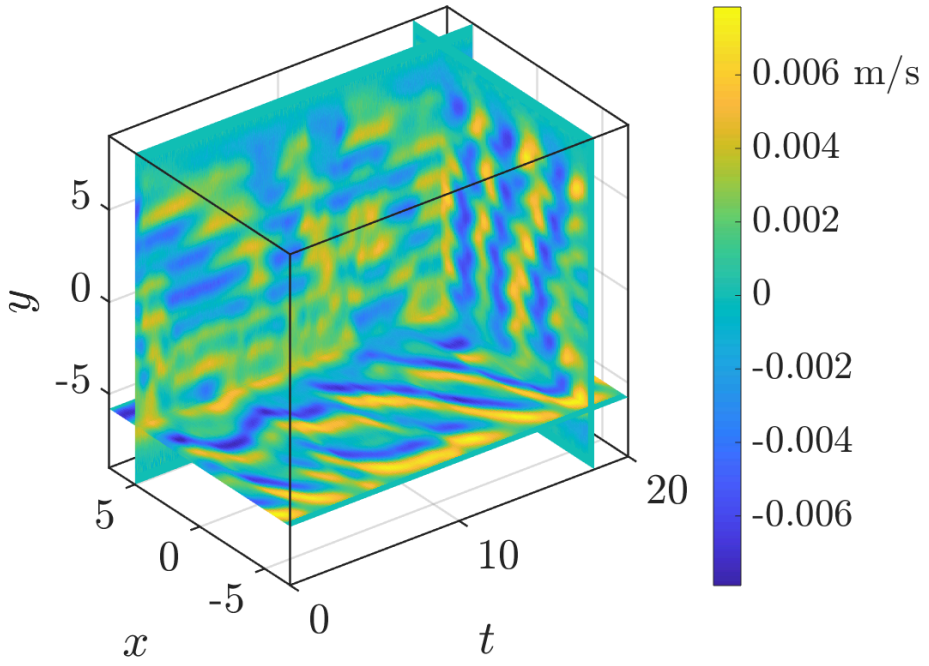
$$\phi_j(x, y, t) = E_\alpha(x)E_\beta(y)E_\gamma(t)P_\lambda(x)P_\mu(y)P_\nu(t), \tag{6.5}$$

where $P_m(x)$ is the m^{th} Legendre polynomial. There are also data sets that are much longer in time that do not require extra sampling with the Legendre polynomials, but they should not receive any detriment by their addition either, so the same sampling methodology was used for each data set. Namely, $I = 50$ unique integration domains were chosen at random, as in Figure 5.5, and for each integration domain λ and μ were varied from 0 and 1, making the total number rows in the library matrix $K = 4I$. The Legendre mode in the time-direction was fixed at $\nu = 1$, since an odd function on $t \in [-1, 1]$ is necessary to satisfy (5.8). Furthermore, since the data is spaced relatively sparsely in space, large envelope exponents (compared to the minimally requried value) are used: $\alpha = \beta = \gamma = 6$.

To sparsify the generalized 2D Navier-Stokes library (2.14), a similar regression procedure as in Section 3.1 is employed again. Again the linear system (1.4) is constructed, and



(a) u



(b) v

Figure 6.3: Representative cross-sections of the experimental horizontal velocity fields acquired via PIV.

the $\mathbf{c}$ is computed as the least squares solution. This $\mathbf{c}$ is then used to calculate the various term norms, $\|c_n \mathbf{q}_n\|$. The key distinction of the weak formulation from the direct approach

Table 6.1: Description of the data sets used for the symbolic regression analysis. Re denotes the mean Reynolds number. Times τ marked with an asterisk (*) represent temporal period, whereas those without represent autocorrelation time.

Re	τ (s)	$\frac{L_t}{\tau}$	$\frac{2H_x}{L_x}$	$\frac{2H_y}{L_y}$	$\frac{2H_t}{L_t}$	$\frac{\Delta x}{w}$	$\frac{\Delta t}{\tau}$
17.88	42*	14	0.80	0.80	0.17	0.15	0.024
17.93	42*	14	0.80	0.80	0.17	0.15	0.024
19.10	42*	14	0.80	0.80	0.17	0.15	0.024
19.75	26	23	0.80	0.80	0.17	0.15	0.039
19.80	28	21	0.80	0.80	0.17	0.15	0.036
22.17	28	128	0.80	0.80	0.028	0.11	0.036
22.27	25	144	0.80	0.80	0.028	0.11	0.040
22.62	24	150	0.80	0.80	0.028	0.11	0.042
23.18	26	138	0.80	0.80	0.028	0.12	0.039
30.88	12	524	0.44	0.46	0.016	0.08	0.083
31.11	13	866	0.48	0.50	0.0089	0.09	0.077
31.26	13	785	0.48	0.50	0.0098	0.09	0.077
35.52	9	2001	0.48	0.50	0.0056	0.09	0.111
35.67	9	2001	0.48	0.50	0.0056	0.09	0.111
36.34	8	149	0.81	0.85	0.083	0.09	0.125

is that the term norms are not compared with the accuracy with which the equations are satisfied (3.5), but rather some small fraction of the magnitude of the time derivative term: $\varepsilon\|\mathbf{q}_0\|$. Since the time derivative is forced to be included by construction (because it is likely to be the dominant term responsible for the time-evolution, by recognizing it as the acceleration in a force-balance equation), it is a natural choice of comparison during the sparsification process. The procedure by which the threshold was chosen is given more consideration below, but $\varepsilon = 0.1$ was found to be a judicious balance between including physically meaningful terms and preventing overfitting.

As usual, to test the method, it was applied to the range of Re shown using ensemble regression; at each Re, $N_{trials} = 30$ different sets of $I = 50$ unique integration domain locations were created and evaluated separately, enabling statistical analysis of the results. The parameter values, as a function of Re, for the terms retained by the procedure are given in Figure 6.4. Incidentally, the identified model is of the same form as that derived from depth-averaging, even though the ansatz used in the depth-averaging procedure is almost

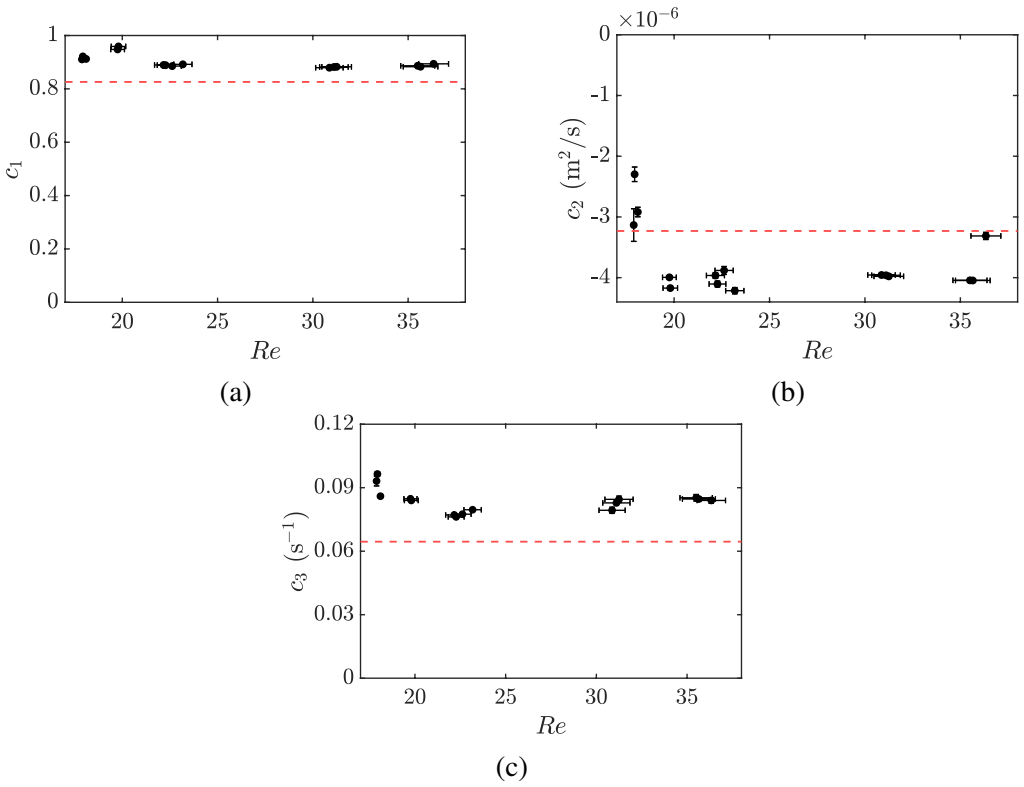


Figure 6.4: Estimated parameters for the discovered model, as a function of Re. The dashed red lines indicate the parameter values computed from the model derived from depth-averaging. The vertical errorbars correspond to the standard deviation in parameter estimation over the ensembles. Similarly, here and below the horizontal errorbars denote standard deviation of $Re(t)$ over the entire trajectory available.

certainly wrong. However, the identified parameter values show significant deviation from their depth-averaged counterparts. All terms, both corresponding to driving (advection) and dispersion (diffusion, linear friction), have their magnitudes amplified from the depth-averaged values: c_1 is about 12% larger, c_2 is around 24% larger, and c_3 can be even 30% larger. That the same model is identified for all data sets, and that the parameter values follow a general trend (ie, they remain more or less constant across Re) is an indicator that the method is giving physically meaningful results. This is especially true when two separate data sets close in Re give close results; consistency isn't a final test, but it is a good first test. Note that the source of the parameter variation across data sets could be due to (the accumulation of) small differences in experimental conditions, as opposed to dynamical parametric dependence on the driving strength.

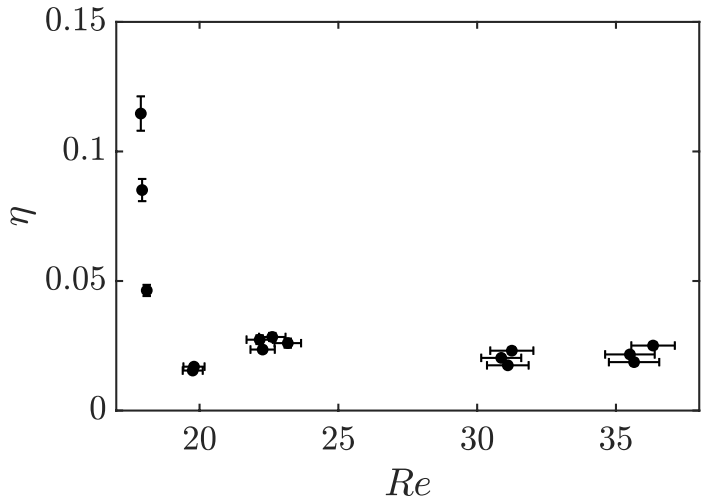


Figure 6.5: Residual η as a function of Re. Note the increased residual for lower Re corresponds to increased standard deviation and deviation from the median for Figure 6.4b.

In addition to the consistency of reconstruction, the method accuracy can also be evaluated based on the residual,

$$\eta \coloneqq \frac{\|Q\mathbf{c} - \mathbf{q}_0\|}{\max_n \|c_n \mathbf{q}_n\|}, \tag{6.6}$$

which is shown as a function of Re in Figure 6.5. This metric can be interpreted as a goodness of fit, and in some sense it measures how well the identified model predicts the flow for the spatiotemporal domains considered. By this metric, the method performs well, as the residual for the chaotic trajectories is below 5%. However, the periodic trajectories show a noticeable deterioration in reconstruction accuracy. Visually inspecting the trajectories reveals that their dynamics are less prominent, ie the magnitude of $\mathbf{u}$ is much smaller than in the chaotic regime. To make this more quantitative, the largest library norm, $\max_n \|c_n \mathbf{q}_n\|_2$, was plotted as a function of Re ($n = 1$, the term corresponding to advection was the largest in all cases). As can be seen in Figure 6.6, the magnitude sharply decreases when transitioning from chaos to periodicity. This decrease is due the fact that only odd Legendre polynomials are used in the weight function, so as to remove the time-independent forcing from the weak form. Hence, as the model terms becomes less and less time-dependent as the system approaches the steady state as it traverses downward in

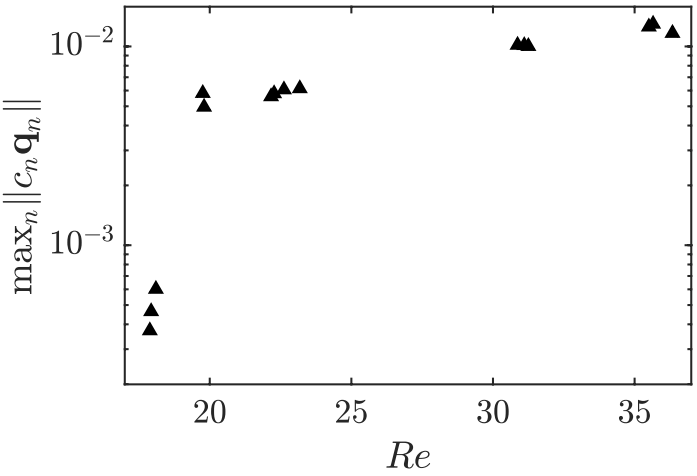


Figure 6.6: The magnitude of the largest library term as a function of Re. The much smaller value in the periodic regime (the leftmost three points) worsens the signal-to-noise ratio, which helps explain the diminished identification accuracy for the periodic trajectories.

Re (the steady-periodic bifurcation occurs very close to $Re = 17.88$), the library norms become smaller and smaller. This then accounts for the decreased reconstruction accuracy observed both in η at low Re, but also in the increase in standard deviation for c_2.

Despite the consistency across Re and $\{\Omega_i\}$ for all the identified terms, it is important to ensure that the retained model terms aren't over-fitting the dynamics. However, knowing which terms are physically relevant is difficult, when not using synthetic data, as the truth isn't known (hence the need for system identification at all). So far, three measures have been provided to bolster the confidence that the identified terms are "real": first, the standard deviation of parameter estimation at a given Re; second, the residual η; and third, the consistency of identification across data sets. All of these lend some assurance that the identified model and its parameters are meaningful, but it is also important to make sure that the threshold ε for truncating the model isn't too high to exclude meaningful terms, or too low to over-fit the data by including terms that aren't physically active. The proportion that the algorithm selects the given term,

$$p_{ID} = \frac{N_{ID}(\varepsilon;\, n)}{N_{trials}}, \tag{6.7}$$

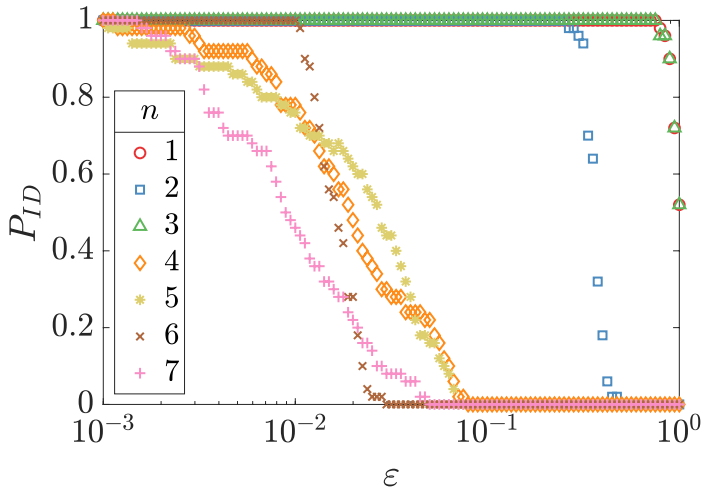


Figure 6.7: The probability of the given term being identified by symbolic regression p_{ID} as a function of the removal threshold ε. For $\varepsilon > 0.1$ the identified model remains consistent.

where N_{ID} is the number of times the model selects the given term and N_{trials} is the number of times the method is applied to the dataset (which corresponds to unique sets of integration domain selections), is shown in Figure 6.7 as a function of ε.

The figure provides information on how large the model constituents are, with respect to the time derivative term. Moving from right to left, at $\varepsilon \approx 1$ advection and Rayleigh friction are included in the model. Interestingly, they appear at the same time, suggesting that these most influential mechanisms balance each other. Then, at $\varepsilon \approx 0.25$ diffusion is included in the model. Unlike the previous two effects added to the model, the Laplacian term appears alone, making it similar to a higher order correction to the dominant balance between advection and the vertical damping effects. Finally, below 10% of $\|\mathbf{q}_0\|_2$, the remaining terms allowed by symmetry begin to be included in the model. However, this is about the error with which the model can be accurately estimated ($\eta \lesssim 0.1$ if normalized by $\|\mathbf{q}_0\|$ instead of $\max_n \|c_n \mathbf{q}_n\|$). So, it becomes unclear as to whether or not these terms are fitting dynamics or over-fitting the noise. On the other hand, if the terms' magnitudes are that small, they may not be so imperative to include in the model to properly understand the dynamics. As a final check, the parameter value for these terms can be tracked across

different choices of integration domains. If the corresponding c_n is kept but varies in sign or in order of magnitude, it indicates that over-fitting is likely. Such was the case for the terms included below $\varepsilon < 0.1$, so this value was chosen as the threshold for the symbolic regression procedure.

6.3 Reconstructing Latent Variables

A model has been identified directly from the data, but it is incomplete. Though being able to find a model in the presence of latent variables is a boon, at its heart the weak formulation accomplishes this by removing the latent variables from consideration altogether. As such, knowledge about c_8 and c_9 in (2.14), or the functional form of (2.15) is lost. However, it turns out that knowing the partial model is enough to mitigate the destructive nature of the weight function by enabling reconstruction of both of the latent variables. First, consider the identified model rearranged as

$$\mathbf{s}(\mathbf{x},t) = \partial_t \mathbf{u} - c_1(\mathbf{u}\cdot\nabla)\mathbf{u} - c_2\nabla^2\mathbf{u} - c_3\mathbf{u} \tag{6.8}$$

$$= c_8\nabla p + c_9\mathbf{f} \tag{6.9}$$

$$= \nabla\phi + \nabla\times\mathbf{a}, \tag{6.10}$$

where the last equality denotes the Helmholtz decomposition of $\mathbf{s}(\mathbf{x},t)$. Subsequently, the forcing vector can also be decomposed into

$$\mathbf{f} = \begin{bmatrix} f(x,y) \\ 0 \end{bmatrix} = \nabla f_\phi + \nabla\times\mathbf{f}_a. \tag{6.11}$$

By considering the divergence of the forcing, some insight gain be gained about its decompositions. Namely, $\nabla\cdot\mathbf{f} = \nabla^2 f_\phi$, hence, if the forcing profile is divergence-free, then that implies that it contains no gradient decomposition. Since the bar magnets are close to uniform, they have translational symmetry in the x direction along the majority of their

length; ie,

$$\partial_x f(x,y) = 0 = \nabla \cdot \mathbf{f},\ \forall x \text{ s.t. } |x| \lesssim 5. \tag{6.12}$$

Furthermore, very far from the magnets the magnetic field should go to zero, which would also make the forcing profiles divergence go to zero. Therefore, the forcing profile should comprise entirely its curl decomposition for the majority of the spatial domain, which in turn gives it one-to-one correspondence with $\mathbf{a}$. This does not hold at the very edge of the magnets, where the field profile transitions from its bulk value to 0; as such, the gradient piece of the Helmholtz decomposition can be expected to contain contributions from the forcing term in this region. However, for most of the domain, where $\nabla \cdot \mathbf{f} = 0$, p is directly proportional to ϕ.

To utilize this information, convert $\mathbf{s}(x,t)$ to Fourier space,

$$\mathbf{S}(\mathbf{k},t) = \iint e^{-i\mathbf{k}\cdot\mathbf{x}}\mathbf{s}(\mathbf{x},t)d\mathbf{x} \tag{6.13}$$

$$= \iint e^{-i\mathbf{k}\cdot\mathbf{x}}\left[\partial_t\mathbf{u} - c_1(\mathbf{u}\cdot\nabla)\mathbf{u} - c_2\nabla^2\mathbf{u} - c_3\mathbf{u}\right]d\mathbf{x} \tag{6.14}$$

$$= i\mathbf{k}\Phi(\mathbf{k},t) + i\mathbf{k}\times\mathbf{A}(\mathbf{k},t), \tag{6.15}$$

where the last equality follows from (6.10). The decomposition fields can be obtained from the following formulas,

$$\Phi = i\frac{\mathbf{k}\cdot\mathbf{S}(\mathbf{k})}{\|\mathbf{k}\|^2}, \tag{6.16}$$

$$\mathbf{A} = i\frac{\mathbf{k}\times\mathbf{S}(\mathbf{k})}{\|\mathbf{k}\|^2}. \tag{6.17}$$

Then, the forcing and pressure (on the inner domain), can be evaluated via

$$p(x,y,t) = \iint e^{i\mathbf{k}\cdot\mathbf{x}}\Phi(\mathbf{k},t)d\mathbf{k}, \tag{6.18}$$

$$f(x,y) = \langle \iint e^{i\mathbf{k}\cdot\mathbf{x}}ik_yA(\mathbf{k},t)d\mathbf{k}\rangle_t, \tag{6.19}$$

where $A(\mathbf{k})$ is the $\hat{z}$ component of $\mathbf{A}(\mathbf{k})$. Evaluating these expressions is done numerically using a Fast Fourier Transform (FFT), where contributions outside the box $\{\mathbf{k} : |k_x| < 2k_0, |k_y| < 2k_0\}$ are thrown away ($k_0 = 2\pi/w$ is the wave number corresponding to the periodicity of the magnet array). Including wave numbers above this yields estimates of $\mathbf{f}(x, y)$ that have unrealistic spatial variations, likely due to observation noise corrupting the high wave number data. Although the spatial derivatives are naturally handled via a FFT, the time derivative in this case is evaluated with a second-order central difference scheme on data smoothed by excluding wave-numbers outside the same box used to take the derivatives. Finally, since reconstructing the forcing profile involves one more derivative than for the pressure, it's fidelity is expected to be worse. This can be counteracted to some degree by the time-averaging in (6.19) on the acquired snapshots, since the forcing profile is expected to be time-independent.

The results of the reconstruction for the forcing profile are shown in Figure 6.8. For the inner region, the fit does well; so much so, that the y-dependence on the center line is indistinguishable between the reconstructed forcing profile and the measured magnetic field. However, near the ends, Gibbs phenomenon begins to develop in the Helmholtz reconstruction; this occurs likely due to the fact that the reconstructing the forcing involves two net derivative evaluations in (6.19), which correspond to the fact that the Laplacian term is present in $\mathbf{s}$. It is for this reason that the result is time-averaged, as the individual time measurements have reduced fidelity.

The other reconstructed field, the pressure, does not have a known field that it can be directly compared against. However, low pressure regions should correspond to high magnitude vorticity regions. In the neighborhood of a pressure minima, the body force $-\nabla p$ is central (in a coordinate system centered at the minima). This central force provides the centripetal acceleration necessary for vortical structures to form. Hence, vortices observed in the experiment should occur at the location of the minima in the reconstructed pressure field. This is exactly what can be seen in Figure 6.9, where snapshots of the vorticity ob-

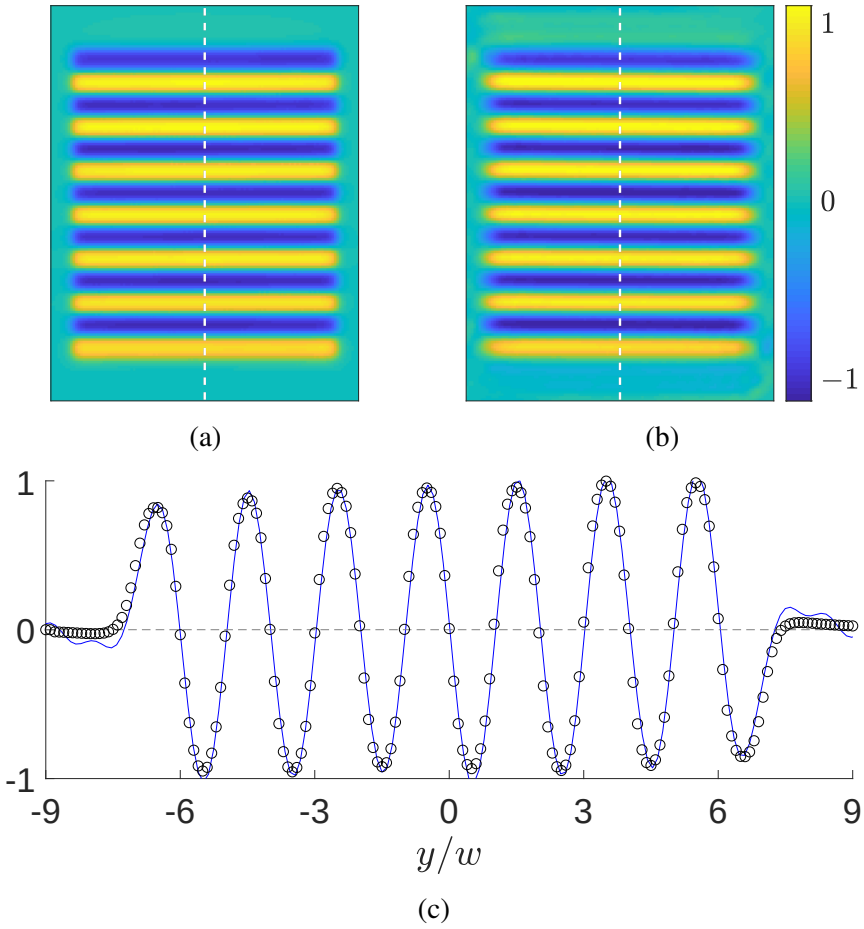


Figure 6.8: Comparison of the reconstructed forcing profile and the measured magnetic field. In (a) the depth-averaged measured magnetic field is shown, while the forcing profile acquired via Helmholtz reconstruction is shown in (b). A vertical cross-section is shown in (c), at the location indicated by the white dashed line in (a) and (b). The reconstructed forcing is in blue and the experimental measurements are black circles. They have been normalized to maximize agreement between their shapes.

servations are compared with the reconstructed pressure at the corresponding times. The good agreement implies that the reconstruction is meaningful.

Using the Helmholtz reconstruction, the removal of the latent variables via the special weight function chosen is counteracted. This reconstruction can then be used to infer information about the model for the pressure variable, whereas the model had to be addressed previously using exclusively physical knowledge, it could be possible to employ more machine learning. One could also imagine other uses for the reconstructions. For example, the reconstructed forcing could be used to simulate the flow if no other options were available

(while in this case a dipole approximation and experimental measurements are available).

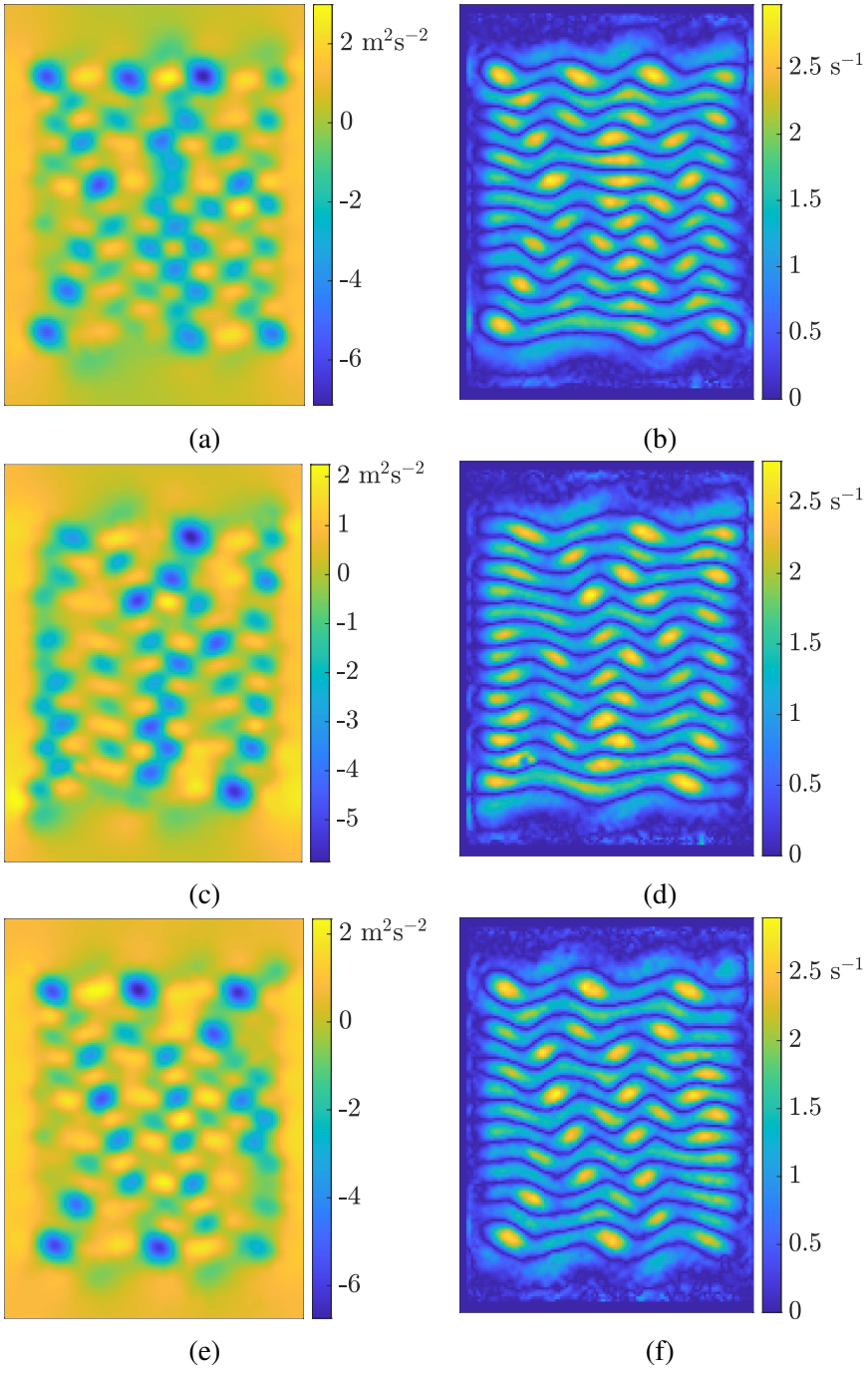


Figure 6.9: The pressure p (left column) compared with magnitude of the vorticity $|\hat{z}\cdot\nabla\times\mathbf{u}|$ (right column) for times $t = 1000$ s, 2000 s, and 3000 s from top to bottom. Observe the correspondence between pressre minima on the left (localized blue regions) and the vortices on the right (localized yellow regions).

CHAPTER 7
CONCLUSION

7.1 Takeaways

Symbolic regression is a way to obtain a model by using direct system observations, rather than derivation from first principles or with phenomenological considerations. This data-driven approach is becoming more feasible as larger amounts of data are becoming accessible, but also more essential as increasingly more complicated systems (for which first principles may not be known, or are difficult to leverage) are attracting the interest of scientists.

However, while symbolic regression has seen success for systems described by ordinary differential equations, applying this approach to spatially distributed systems described by partial differential equations proved more difficult. Measurements of such systems have intrinsic issues, namely, they are spatially and temporally discrete, and they are corrupted by measurement noise. Furthermore, some variables necessary to fully describe the evolution of the system may not be observable at all. Additionally, it is not always clear which variables are important and how they should appear in a model. The work presented here addresses these concerns inherent to high-dimensional spatially-distributed systems.

Before any regression can be performed, the relevant variables of the system must be identified. As illustrated here, this can be done via the utilization of the available domain knowledge, such as the relevant (known) physics. For example, for a fluid flow we expect a dynamical model to describe the change in the momentum of the fluid in response to the external driving forces and internal stresses, with the fluid velocity and pressure being the relevant variables. After relevant variables are identified, additional domain knowledge can be used to reduce them to a tractable set. Again using thin fluid flow as an example,

knowing that the fluid motion is largely restricted to the horizontal direction means we may not need not consider variables corresponding to the vertical direction (the vertical velocity, or the vertical component of the external forces such as gravity). Similar considerations are often used to construct first-principles models.

The next step is to obtain equations governing the relevant variables, which can be in the form of evolution equations or constraints. For a fluid flow, examples of the former and the latter are the Navier-Stokes equation and the incompressibility condition. Construction of these equations can again be guided by domain knowledge, such as symmetry considerations. For example, the physics governing the horizontal fluid flow is invariant with respect to spatial translations in the horizontal directions, so we expect the model to have no explicit dependence on position. Furthermore, at the macroscopic scale the fluid can be considered continuous, hence we expect the model to be represented by one or more PDEs with constant coefficients (aside from external forcing, which may break the symmetry). Also, if the physics is invariant with respect to rotations, every term in the model terms should transform identically under rotation. For instance, the evolution equation for the momentum of the fluid – a vector field – should only contain terms that transform as vectors under rotation. The available domain knowledge constrains, but does not determine the form of the model. For instance, one can construct an infinite number of terms that transform as vectors using vector objects such as the fluid velocity, external forces or the gradient operator and scalar objects such as pressure or a scalar product of vector objects. The models therefore have to be truncated using other considerations, such as the degree of nonlinearity or the highest order of spatial or temporal derivatives.

The hybrid approach combining data-driven aspects with relevant domain knowledge is in stark contrast to purely data-driven approaches such as that used by Rudy et. al [20] who built candidate models out of terms in the form of arbitrary polynomial combinations of the relevant fields and spatial derivatives. For instance, their model for vorticity of a two-dimensional fluid flow included the terms $\partial_x^2\omega$ and $\partial_y^2\omega$ independently, with different

coefficients, while rotational symmetry requires them to enter in the combination $\nabla^2\omega$, i.e., have the same coefficient. Utilizing a hybrid approach as presented in this thesis removes terms inconsistent with the known physics ahead of time; while symbolic regression may be able to eliminate such terms, including them can make the model identification process both computationally expensive and numerically unstable. The hybrid approach also improves interpretability of the resulting model; for example, while the terms $u\partial_x u$, $u\partial_x v$, $v\partial_y u$, $v\partial_y v$ in the model of a fluid flow may all be identified independently, they will have similar, but different, coefficients. This obscures the true form of the model which contains the advection term $(\mathbf{u}\cdot\nabla)\mathbf{u}$.

The hybrid approach is independent of the implementation of symbolic regression – it only helps construct a library of candidate models that the symbolic regression procedure narrows down to the parsimonious form. Though it offers a method of preparing for the regression, it does nothing to address the three key problems (discreteness of the data, the presence of noise and, potentially, latent variables) that complicates the application of symbolic regression to spatially distributed systems. This is illustrated by the failure of symbolic regression applied to the strong formulation of the model as discussed in Ch. 3. This shortcoming is not limited to the polynomial interpolation. Any approach that relies on local evaluation of the model terms (e.g., using derivatives estimated through spectral methods, finite differences, or regularization methods such as Total Variation, etc.) will be fundamentally limited in its accuracy. The accuracy of any derivative estimate, especially higher order derivatives, is crucially dependent on both the noise corruption and the grid density. The presence of latent variables exacerbates the problem; while dependence on the latent variables can be eliminated by applying a differential operator, this raises the order of the governing equation(s) further decreasing the accuracy of the reconstructed model.

On the other hand, by considering the weak formulation of a candidate model (i.e., by applying an integral operator rather than a differential operator), many of these problems can be avoided altogether. For terms of the appropriate type, the derivatives present

in the model terms can be transferred onto a continuous and noiseless weight function. To illustrate the magnitude of the improvement, recall that, in the strong formulation, the Kuramoto-Sivashinsky equation was identified with a 50% parameter error from data with only 0.1% noise corruption [20]. On the other hand, in the weak formulation, the parameters could be identified with less than 1% error for noise corruption up to 10%. In the former case, symbolic regression essentially failed to identify the correct model despite the extremely low level of noise, while in the latter case, the correct model was identified using data with noise level that is comparable to, or even higher than, what one might expect for experimental measurements.

The benefits to using a weak formulation also extends to the (lack of) sensitivity of the results to the sparsity of the grid on which the data is acquired. Take for example DNS of the Kolmogorov flow. For strong formulation used in Ch. 3, a density of 40 grid points per correlation length was necessary to produce parameter error less than 1%, while using a density of 10 grid points per correlation length in Ch. 5 was sufficient to achieve the same accuracy using the weak formulation. Furthermore, the sensitivity of the approach to the grid spacing is controllable by choice of hyper-parameters of the weak formulation. The scaling of the parameter error with grid spacing and "envelope exponent" was demonstrated in both Ch. 4 and Ch. 5. The main takeaway from those investigations is that the negative effects of having a low grid resolution can be offset by making more of the weight function's derivatives continuous (i.e.., vanishing) at the boundaries of the integration domains.

Though the weak formulation doesn't require a fine grid, it does require information from a larger domain than a corresponding strong formulation. However, this can be offset by sampling the same domain with different weight functions; choosing the weight function as a product of an envelope function that satisfies the boundary conditions and another (near) arbitrary function makes doing this easy. The weight function can also be designed to remove latent variables from the weak form of the model, with minimal effects on the

performance. For example, a model for the Kolmogorov flow was identified without any knowledge of the pressure or forcing profile; to accomplish this same task using the strong formulation, a derivative operator was used that severely exacerbated the noise sensitivity. Enforcing odd parity of the weight function in the time direction of the integration domain does restrict the space of allowable weight functions that can be used to acquire unique library rows, but this is a far smaller price to pay than raising the order of the governing equation.

After constructing a library of candidate models using relevant domain knowledge, and numerically evaluating the integral of each term, a regression method is necessary to discern which terms should be included in a model that is both simple accurately reproduces the data. The presented method is a slight variation on the existing sequentially thresholded least squares algorithm which lies at the core of SINDy [9]. The issue with this is that the choice of threshold is not known *a priori*, but this thesis takes steps in providing a procedure by which a good choice of threshold can be determined when the true model is unknown. Using ensemble analysis and judging the variation of the coefficients might provide a way to discern between terms included via overfitting, and which should actually be present. For instance, for experimental fluid flow data considered in Chapter 6, a threshold was determined by looking at the variation in the magnitude or sign of the coefficients across an ensemble of regression results.

Being able to get a consistent model for experimental data with a coarse grid, noise corruption, and with missing information is great, but ultimately the missing information contained in the latent variables was circumvented by special choice of weight function. Though it is a definite improvement from local approaches, losing the information contained in the latent variables isn't preferable. However, by utilizing the model identified via symbolic regression, the pieces thrown away by our specially chosen weight function can be recreated. For example, the components of the Helmholtz decomposition for the experimental fluid flow have very clear correspondence to the pressure and Lorenz forcing.

A direct relationship between Helmholtz components and latent variables, as here, is not generally the case.

It should be mentioned that, although the presented approach was only tested for models in the form of (nonlinear) PDEs, it should also apply to algebraic equations, ODEs, integro-differential equations, delay-differential equations, and systems of such equations. The approach does have some limitations, though.

Unfortunately, utilizing domain knowledge and considering the weak form of the governing equations isn't a silver bullet that enables model discovery in all cases. First, not all potential terms can have their derivatives transferred onto the weight function w. For example, there is no way to re-write the term

$$\int w\partial_x v\partial_t u d\Omega,$$

where there are no derivatives on the observed variables, u and v. In this case these derivatives are unavoidable, but in some cases the order of derivatives can be reduced, such as a term like

$$\int wu\partial_x^2 u d\Omega = -\int \partial_x(wu)\partial_x u d\Omega,$$

could have the necessary number of derivatives reduced (here, from second to first). This is the case for the candidate models of Kolmogorov flow discussed in Ch. 6. In such cases it is inevitable that the aforementioned methods to evaluate derivatives be returned to. This will provide an accuracy floor to the method likely higher than what it would have been without having to evaluate these derivatives, but the method should not altogether break down unless higher-order derivatives are unavoidable. Fortunately, terms that are both highly nonlinear and involve high-order derivatives appear quite rarely in models of natural phenomena. How the local errors in state and derivative estimation affect the error of the global integral is not immediately obvious either. Understanding precisely how the nonlinear noise corruption affects the value of the integral is a potential investigation

direction.

Models with arbitrarily complex parametric dependence would also prove difficult for a weak formulation approach to handle, as such dependence would not be easily represented by the linear combination of some basis functions, as in Section 4.4. More model forms that could prove troublesome are those that involve latent variables without exploitable symmetry. The removal of the pressure and stationary forcing, though useful, was only a special case and shouldn't be expected to be generically possible.

The weak formulation also imposes certain conditions on the type of data it needs. For the integrals to be accurate, the data needs to be available on non-local domains. In the quasi-2D flow system, the integration domains needed to occupy around half or more of the spatial domain to provide accurate results. Making many measurements over a large region may not always be feasible.

7.2 Future Work

Only candidate models in the form of a linear superposition of terms (possibly nonlinear in the dependent variables) were considered here, with unknown parameters serving as the coefficients. Needless to say, this is not the most general form of a model. For instance, the dynamics of biological and chemical systems is often described by models where unknown parameters enter in a more complicated fashion. Limited progress has been made in identifying models where parameters enter in a nonlinear way, for both ODEs [35] and algebraic equations [36]. Whether a similar approach can be used for models of spatially extended systems remains unclear due to both conceptual issues (much richer library of terms and operations) and the associated computational cost (curse of dimensionality).

The choice of the weight functions has not been explored systematically, aside from their effect on the accuracy with which numerical quadratures are evaluated [34]. In most cases here, a large amount of data was available, and so it was not necessary to squeeze out as much information as possible from the data set and a single weight function was

sufficient to recover a parsimonious model with sufficient accuracy. However, in practice the amount of available data can be quite limited. In this case, filling out the rows of the library matrix by using a large number of integration domains isn't possible. The remaining option is to sample the data within the same integration domain with different weight functions. In the work presented here, using multiple weight functions led to less accurate results than using different integration domains. Understanding why that is, and moreover to optimize the weight functions to extract the most information from a small data set is also an interesting direction of inquiry. Many weight functions have been used, and they are all qualitatively similar [30, 26, 4, 33], but a systematic comparison and optimal choice has not been shown.

There are systems whose models involve terms of mundane form (that can be in principle handled with the weak formulation), but whose dynamics pose practical problems. For example, the director field $\mathbf{n}(x, y, t)$ describing nematic active matter suspended in a fluid layer is prone to formation of topological defects. Another example is the Burgers equation that tends to develop shocks, where solution is (nearly) discontinuous. Spatial (and temporal) derivatives at the location of the defects diverge and, unless the regions near the defects are isolated, skew the results of symbolic regression. Indeed, PDE models of such systems are designed to describe the dynamics away from defects and their discretization near defects will be inconsistent with discretely sampled data. This problem can be addressed by designing the weight function to mask the defects by vanishing in their neighborhoods. Such weight functions can even be designed to track the defects automatically by making them data-dependent.

Finally, a few comments on the form of the linear system whose sparse solution defines a parsimonious PDE model are in order. In this work, the linear system (1.4) with $\mathbf{q}_0 \neq 0$ was constructed by separating out a time-derivative term such as $F_0 = \partial_t^k u$, but this need not be the case. Here the choice was made because a time-derivative term is always present in an evolution equation. However, not only will some PDEs not include a time-derivative

(incompressibility condition for liquids is one example), even if present it may not be the best choice. For instance, if the data set describes a steady state, this term will vanish identically, yielding the trivial least squares solution $\mathbf{c} = 0$. One solution is to collect all the terms on the same side of the equation, yielding the linear system

$$\hat{Q}\mathbf{c} = 0.$$

Nontrivial solutions to this equation are given by the singular vectors of $\hat{Q}$ corresponding to zero singular values. In practice, this equation won't be satisfied exactly, but choosing the lowest singular value is a sensible choice. This works well sometimes, for instance the analysis done in Chapter 4 was also completed using the SVD approach [34]. However, in other cases the lowest singular value did not correspond to reasonable parameter values, even though the solutions to (1.4) were well-defined. Identifying the root cause of this discrepancy and learning how to avoid it in the SVD approach could yield a more general way of handling the regression procedure.

The field of model discovery is experiencing a large amount of growth, with data being ever more available, machine learning techniques being advanced, and systems of interest becoming less susceptible to first-principles modeling due to their complexity. While a lot of the popular methods have proven ineffective for spatially distributed systems, the weak formulation/method function approach has been shown here to provide an excellent and highly flexible alternative. Besides refining models for systems that are relatively well understood qualitatively, like for the Kolmogorov flow in the moderately high Re regime, or for systems in which a large library has been proposed but ill-refined for particular scenarios, like the active nematics, opportunity exists to discover entirely new physics or biology that first principles would have a difficult time with. Hopefully, the work presented here will aid the discovery of physics for extended systems that would otherwise remain quantitatively misunderstood.

Appendices

APPENDIX A

GUIDE TO THE MATLAB SYMBOLIC REGRESSION CODE FOR 2D FLOWS

The codes necessary to perform symbolic regression on 2D flows were originally developed around the constructing matrices for input into the SINDy algorithm [9]. Processing the data prior to this step takes up the bulk of the code, and everything but the SINDy algorithm (which itself is modified) was built independently. More tips and hints are located within the README.txt that accompanies the codes. Before using the main code, it is necessary to get the velocities into the correct form. The x-velocity should be saved as U_t, and the y-velocity as V_t, where the first dimension corresponds to y, the second dimension corresponds to x, and the third dimension corresponds to t. In addition, their spatial derivatives, if needed, should be saved in the format dxU_t. The grid spacing should also be saved as dx, dy, and dt. All quantities should be in mks units. Finally, it is important to save the .mat file with the '-v7.3' flag.

`SymbReg_WF_Kolmo()`

- Description: The main code. It takes a trajectory and hyperparameters as input, and outputs a model to describe the trajectory.
- Inputs:
    1. filename – (string) file name that contains velocities, their spatial derivatives, and grid parameters.
    2. term_names – (string array) contains codes that tells which terms to populate the library with
    3. N_d – (int) the number of integration domains to sample
    4. N_h – (3x1 int array) max weight function mode numbers in x, y, t, respectively

5. D – (3x1 double array) the size of the integration in x, y, t, respectively
6. env_exp – (3x1 int array) the envelope exponents in x, y, t, respectively
7. gamma – (double) sparse regression threshold
8. track – (bool) whether or not to track progress in the command window
9. vis – (bool) whether or not to visualize the first set of integrands

- Outputs:

1. coeffs – (struct) contains the identified coefficients with their corresponding term code
2. nrms – (1xN double array) contains the relative term-norms of the retained model terms
3. res – (double) relative residual
4. Q – (KxN+1 double array) library matrix (including time-derivative)
5. c – (N+1x1 double array) coefficient vector (including time-derivative)
6. P – (N_dx3 double array) locations of the integration domains used to sample
7. figInterlock – (figure) figure object if vis = 0

`computeTikhonovDerivatives()`

- Description: Uses Tikhonov regularization to take the first-order derivatives of the horizontal velocities.

- Inputs:

1. fileName – (string) reference to .mat file that contains the velocities you want the derivatives of
2. take_x – (bool) whether or not to take x-derivatives
3. take_y – (bool) whether or not to take y-derivatives

4. take_y – (bool) whether or not to take t-derivatives
5. save_along – (bool) whether or not to periodically save the results

`defineLibraryKolmo()`

- Description: Creates the necessary strings in order to evaluate the integrals of specified terms
- Inputs:
  1. term_names – (string array) contains the term codes to specify which terms to populate the library with
- Outputs:
  1. lib_names – (string array) contains the term codes to specify which terms to populate the library with, but ordered such that u_t is first
  2. krnl_str – (container map) each string in lib_names maps to a string in the container that provides instructions on how to evaluate the corresponding integrals
  3. tex_names – (string array) contains names of terms in LaTeX format

`glossaryKolmo()`

- Description: Provides list of term codes and their corresponding descriptions
- Output:
  1. glossary – (string array) contains all possible term codes that can be used in `SymbReg_WF_Kolmo()` and their descriptions

`SINDy()`

- Description: Performs sparse regression
- Inputs:

1. Q – (KxN double array) library matrix
2. q0 – (Kx1 double array) time-derivative column
3. gamma – (double) sparsification threshold

- Outputs:

1. c – (Nx1 double array) coefficient vector

REFERENCES

[1] T. Hall, *Carl Friedrich Gauss, A Biography*. MIT Press, 1970.

[2] W. K. Buhler, *Gauss: A Biographical Study*. Springer-Verlag, 1981.

[3] M. Shinbrot, "On the analysis of linear and nonlinear dynamical systems from transient-response data," 1954.

[4] F. J. Perdreauville and R. Goodson, "Identification of systems described by partial differential equations," 1966.

[5] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[6] G. L. Baker, J. P. Gollub, and J. A. Blackburn, "Inverting chaos: Extracting system parameters from experimental data," *Chaos*, vol. 6, no. 4, pp. 528–533, 1996.

[7] D. P. Vallette, G. Jacobs, and J. P. Gollub, "Oscillations and spatiotemporal chaos of one-dimensional fluid fronts," *Physical Review E*, vol. 55, no. 4, p. 4274, 1997.

[8] M. Bär, R. Hegger, and H. Kantz, "Fitting partial differential equations to space-time dynamics," *Physical Review E*, vol. 59, no. 1, p. 337, 1999.

[9] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[10] D. Xu and O. Khanmohamadi, "Spatiotemporal system reconstruction using fourier spectral operators and structure selection techniques," *Chaos*, vol. 18, no. 4, p. 043 122, 2008.

[11] H. Schaeffer and S. G. McCalla, "Sparse model selection via integral terms," *Physical Review E*, vol. 96, no. 2, p. 023 302, 2017.

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[13] H. Schaeffer, "Learning partial differential equations via data discovery and sparse optimization," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2197, p. 20 160 446, 2017.

[14] X. Li, L. Li, Z. Yue, X. Tang, H. U. Voss, J. Kurths, and Y. Yuan, "Sparse learning of partial differential equations with structured dictionary matrix," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 4, p. 043 130, 2019.

[15] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[16] L. Zhang and H. Schaeffer, "On the convergence of the sindy algorithm," *Multiscale Modeling & Simulation*, vol. 17, no. 3, pp. 948–972, 2019.

[17] A. A. R. AlMomani, J. Sun, and E. Bollt, "How entropic regression beats the outliers problem in nonlinear system identification," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 1, p. 013 107, 2020.

[18] T. Müller and J. Timmer, "Parameter identification techniques for partial differential equations," *International Journal of Bifurcation and Chaos*, vol. 14, no. 06, pp. 2053–2060, 2004.

[19] O. Khanmohamadi and D. Xu, "Spatiotemporal system identification on nonperiodic domains using chebyshev spectral operators and system reduction algorithms," *Chaos*, vol. 19, no. 3, p. 033 117, 2009.

[20] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, e1602614, 2017.

[21] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[22] A. N. Tikhonov, "On the solution of ill-posed problems and the method of regularization," in *Doklady Akademii Nauk*, Russian Academy of Sciences, vol. 151, 1963, pp. 501–504.

[23] J. Cullum, "Numerical differentiation and regularization," *SIAM Journal on numerical analysis*, vol. 8, no. 2, pp. 254–265, 1971.

[24] I. Knowles and R. J. Renka, "Methods for numerical differentiation of noisy data," *Electron. J. Differ. Equ*, vol. 21, pp. 235–246, 2014.

[25] R. Chartrand, "Numerical differentiation of noisy, nonsmooth data," *ISRN Applied Mathematics*, vol. 2011, 2011.

[26] K Takaya, "The use of hermite functions for system identification," *IEEE Transactions on Automatic Control*, vol. 13, no. 4, pp. 446–447, 1968.

[27] T. Co and B. Ydstie, "System identification using modulating functions and fast fourier transforms," *Computers & chemical engineering*, vol. 14, no. 10, pp. 1051–1066, 1990.

[28] S. Jalali, J. Jordan, and R. Mackie, "Measurement of the parameters of all-pole transfer functions using shifted hermite modulating functions," *Automatica*, vol. 28, no. 3, pp. 613–616, 1992.

[29] H. Preisig and D. Rippin, "Theory and application of the modulating function method—iii. application to industrial process, a well-stirred tank reactor," *Computers & chemical engineering*, vol. 17, no. 1, pp. 29–39, 1993.

[30] ——, "Theory and application of the modulating function method—i. review and theory of the method and theory of the spline-type modulating functions," *Computers & chemical engineering*, vol. 17, no. 1, pp. 1–16, 1993.

[31] J. Rudolph and F. Woittennek, "An algebraic approach to parameter identification in linear infinite dimensional systems," in *2008 16th Mediterranean Conference on Control and Automation*, IEEE, 2008, pp. 332–337.

[32] N. Gehring, T. Knüppel, J. Rudolph, and F. Woittennek, "Algebraic identification of heavy rope parameters," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 161–166, 2012.

[33] S. Asiri and T.-M. Laleg-Kirati, "Modulating functions-based method for parameters and source estimation in one-dimensional partial differential equations," *Inverse Problems in Science and Engineering*, vol. 25, no. 8, pp. 1191–1215, 2017.

[34] D. R. Gurevich, P. A. Reinbold, and R. O. Grigoriev, "Robust and optimal sparse regression for nonlinear pde models," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 10, p. 103 113, 2019.

[35] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 104, no. 24, pp. 9943–9948, 2007.

[36] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *science*, vol. 324, no. 5923, pp. 81–85, 2009.

[37] L. Breiman and J. H. Friedman, "Estimating optimal transformations for multiple regression and correlation," *Journal of the American statistical Association*, vol. 80, no. 391, pp. 580–598, 1985.

[38] J Timmer, H Rust, W Horbelt, and H. Voss, "Parametric, nonparametric and parametric modelling of a chaotic circuit time series," *Physics Letters A*, vol. 274, no. 3-4, pp. 123–134, 2000.

[39] H Voss, M. Bünner, and M. Abel, "Identification of continuous, spatiotemporal systems," *Physical Review E*, vol. 57, no. 3, p. 2820, 1998.

[40] H. U. Voss, P. Kolodner, M. Abel, and J. Kurths, "Amplitude equations from spatiotemporal binary-fluid convection data," *Physical review letters*, vol. 83, no. 17, p. 3422, 1999.

[41] M. Ghil and P. Malanotte-Rizzoli, "Data assimilation in meteorology and oceanography," in *Advances in geophysics*, vol. 33, Elsevier, 1991, pp. 141–266.

[42] N. K. Nichols, "Mathematical concepts of data assimilation," in *Data assimilation*, Springer, 2010, pp. 13–39.

[43] P. Smith, G. Thornhill, S. Dance, A. Lawless, D. Mason, and N. Nichols, "Data assimilation for state and parameter estimation: Application to morphodynamic modelling," *Quarterly Journal of the Royal Meteorological Society*, vol. 139, no. 671, pp. 314–327, 2013.

[44] J. Seinfeld, "Identification of parameters in partial differential equations," *Chemical engineering science*, vol. 24, no. 1, pp. 65–74, 1969.

[45] T. G. Müller and J. Timmer, "Fitting parameters in partial differential equations from partially observed noisy data," *Physica D: Nonlinear Phenomena*, vol. 171, no. 1-2, pp. 1–7, 2002.

[46] G Chavent, R Goodson, and M Polis, "Identification of parameter distributed systems," pp. 31–48, 1974.

[47] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *Journal of Computational Physics*, vol. 357, pp. 125–141, 2018.

[48] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Numerical gaussian processes for time-dependent and nonlinear partial differential equations," *SIAM Journal on Scientific Computing*, vol. 40, no. 1, A172–A198, 2018.

[49] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[50] Z. Long, Y. Lu, X. Ma, and B. Dong, "Pde-net: Learning pdes from data," in *International Conference on Machine Learning*, 2018, pp. 3208–3216.

[51] M. Cross and H. Greenside, *Pattern formation and dynamics in nonequilibrium systems*. Cambridge University Press, 2009.

[52] R. A. Fisher, "The wave of advance of advantageous genes," *Annals of eugenics*, vol. 7, no. 4, pp. 355–369, 1937.

[53] A. Kolmogorov, I. Petrovskii, and N. Piskunov, "Study of a diffusion equation that is related to the growth of a quality of matter, and its application to a biological problem," *Byul. Mosk. Gos. Univ. Ser. A Mat. Mekh*, vol. 1, no. 1, p. 26, 1937.

[54] N. Bacaër, *A short history of mathematical population dynamics*. Springer Science & Business Media, 2011.

[55] A. Lemarchand, H. Lemarchand, E. Sulpice, and M. Mareschal, "Microscopic simulation of a wave front: Fluctuation effects on propagation velocity and width," *Physica A Statistical Mechanics and its Applications*, vol. 188, pp. 277–283, 1992.

[56] S. Coen, M. Tlidi, P. Emplit, and M. Haelterman, "Convection versus dispersion in optical bistability," *Physical review letters*, vol. 83, no. 12, p. 2328, 1999.

[57] T. J. Newman, E. B. Kolomeisky, and J. Antonovics, "Population dynamics with global regulation: The conserved fisher equation," *Phys. Rev. Lett.*, vol. 92, p. 228 103, 22 2004.

[58] R. FitzHugh, "Mathematical models of threshold phenomena in the nerve membrane," *The bulletin of mathematical biophysics*, vol. 17, no. 4, pp. 257–278, 1955.

[59] ——, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical journal*, vol. 1, no. 6, pp. 445–466, 1961.

[60] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2061–2070, 1962.

[61] S. Barland, O. Piro, M. Giudici, J. R. Tredicce, and S. Balle, "Experimental evidence of van der pol–fitzhugh–nagumo dynamics in semiconductor optical amplifiers," *Phys. Rev. E*, vol. 68, p. 036 209, 3 2003.

[62] M. Ciszak, S. Euzzor, F. T. Arecchi, and R. Meucci, "Experimental study of firing death in a network of chaotic fitzhugh-nagumo neurons," *Phys. Rev. E*, vol. 87, p. 022 919, 2 2013.

[63] R. R. Aliev and A. V. Panfilov, "A simple two-variable model of cardiac excitation," *Chaos, Solitons & Fractals*, vol. 7, no. 3, pp. 293–301, 1996.

[64] J. Dold, R. Thatcher, A Omon-Arancibia, and J Redman, "From one-step to chain-branching premixed flame asymptotics," *Proceedings of the Combustion Institute*, vol. 29, no. 2, pp. 1519–1526, 2002.

[65] M. Flicker and J. Ross, "Mechanism of chemical instability for periodic precipitation phenomena," *The Journal of Chemical Physics*, vol. 60, no. 9, pp. 3458–3465, 1974.

[66] D Feinn, P Ortoleva, W Scalf, S Schmidt, and M Wolff, "Spontaneous pattern formation in precipitating systems," *The Journal of Chemical Physics*, vol. 69, no. 1, pp. 27–39, 1978.

[67] J. D. Murray, "A pre-pattern formation mechanism for animal coat markings," *Journal of Theoretical Biology*, vol. 88, no. 1, pp. 161–199, 1981.

[68] M. B. Short, M. R. D'orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes, "A statistical model of criminal behavior," *Mathematical Models and Methods in Applied Sciences*, vol. 18, no. supp01, pp. 1249–1267, 2008.

[69] C. Pan, B. Li, C. Wang, Y. Zhang, N. Geldner, L. Wang, and A. L. Bertozzi, "Crime modeling with truncated lévy flights for residential burglary models," *Mathematical Models and Methods in Applied Sciences*, vol. 28, no. 09, pp. 1857–1880, 2018.

[70] A Lefebvre, C Ellien, D Davoult, E Thiébaut, and J. Salomon, "Pelagic dispersal of the brittle-star ophiothrix fragilis larvae in a megatidal area (english channel, france) examined using an advection/diffusion model," *Estuarine, Coastal and Shelf Science*, vol. 57, no. 3, pp. 421–433, 2003.

[71] C. M. Topaz, A. L. Bertozzi, and M. A. Lewis, "A nonlocal continuum model for biological aggregation," *Bulletin of mathematical biology*, vol. 68, no. 7, p. 1601, 2006.

[72] X. Chen, R. Hambrock, and Y. Lou, "Evolution of conditional dispersal: A reaction–diffusion–advection model," *Journal of mathematical biology*, vol. 57, no. 3, pp. 361–386, 2008.

[73] M. Seidel, J. Graue, B. Engelen, J. Köster, H. Sass, and J. Rullkötter, "Advection and diffusion determine vertical distribution of microbial communities in intertidal sediments as revealed by combined biogeochemical and molecular biological analysis," *Organic geochemistry*, vol. 52, pp. 114–129, 2012.

[74] H. Bateman, "Some recent researches on the motion of fluids," *Monthly Weather Review*, vol. 43, no. 4, pp. 163–170, 1915.

[75] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," in *Advances in applied mechanics*, vol. 1, Elsevier, 1948, pp. 171–199.

[76] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics," *Quarterly of applied mathematics*, vol. 9, no. 3, pp. 225–236, 1951.

[77] L. Romagnani, S. V. Bulanov, M. Borghesi, P. Audebert, J. C. Gauthier, K. Löwenbrück, A. J. Mackinnon, P. Patel, G. Pretzler, T. Toncian, and O. Willi, "Observation of collisionless shocks in laser-plasma experiments," *Phys. Rev. Lett.*, vol. 101, p. 025 004, 2 2008.

[78] Y Nakamura, "Experiments on ion-acoustic shock waves in a dusty plasma," *Physics of Plasmas*, vol. 9, no. 2, pp. 440–445, 2002.

[79] C. S. West and T. A. B. Kennedy, "Optical multiwave mixing: Dark solitary wave trains and quasiperiodic dynamics," *Phys. Rev. A*, vol. 47, pp. 1252–1262, 2 1993.

[80] S. Trillo, S. Wabnitz, and T. A. B. Kennedy, "Nonlinear dynamics of dual-frequency-pumped multiwave mixing in optical fibers," *Phys. Rev. A*, vol. 50, pp. 1732–1747, 2 1994.

[81] B. J. Eggleton, C. M. de Sterke, and R. Slusher, "Bragg solitons in the nonlinear schrödinger limit: Experiment and theory," *JOSA B*, vol. 16, no. 4, pp. 587–599, 1999.

[82] S. Trillo and A. Valiani, "Hydrodynamic instability of multiple four-wave mixing," *Optics letters*, vol. 35, no. 23, pp. 3967–3969, 2010.

[83] J. Fatome, C. Finot, G. Millot, A. Armaroli, and S. Trillo, "Observation of optical undular bores in multiple four-wave mixing," *Physical Review X*, vol. 4, no. 2, p. 021 022, 2014.

[84] A. Tikan, C. Billet, G. El, A. Tovbis, M. Bertola, T. Sylvestre, F. Gustave, S. Randoux, G. Genty, P. Suret, and J. M. Dudley, "Universality of the peregrine soliton in the focusing dynamics of the cubic nonlinear schrödinger equation," *Phys. Rev. Lett.*, vol. 119, p. 033 901, 3 2017.

[85] N. N. Akhmediev and A. Ankiewicz, *Solitons: nonlinear pulses and beams*. Chapman & Hall, 1997.

[86] B. M. Lake, H. C. Yuen, H. Rungaldier, and W. E. Ferguson, "Nonlinear deep-water waves: Theory and experiment. part 2. evolution of a continuous wave train," *Journal of Fluid Mechanics*, vol. 83, no. 1, pp. 49–74, 1977.

[87] J. Boussinesq, *Essai sur la théorie des eaux courantes*. Impr. nationale, 1877.

[88] D. J. Korteweg and G. De Vries, "Xli. on the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 39, no. 240, pp. 422–443, 1895.

[89] A. D. Polyanin and V. F. Zaitsev, *Handbook of nonlinear partial differential equations*. Chapman and Hall/CRC, 2016.

[90] N. Bondarenko, E. Gak, and M. Gak, "Application of mhd effects in electrolytes for modeling vortex processes in natural phenomena and in solving engineering-physical problems," *Journal of engineering physics and thermophysics*, vol. 75, no. 5, pp. 1234–1247, 2002.

[91] B. Suri, J. Tithof, R. Mitchell, R. O. Grigoriev, and M. F. Schatz, "Velocity profile in a two-layer Kolmogorov-like flow," *Phys. Fluids*, vol. 26, no. 5, 053601, p. 053 601, 2014.

[92] J. Tithof, B. Suri, R. K. Pallantla, R. O. Grigoriev, and M. F. Schatz, "Bifurcations in a quasi-two-dimensional Kolmogorov-like flow," *Journal of Fluid Mechanics.*, vol. 828, pp. 837–866, 2017.

[93] R. Pallantla, "Exact coherent structures and dynamical connections in a quasi 2d kolmogorov like flow," Ph.D. dissertation, Georgia Institute of Technology, 2018.

[94] B. Suri, J. Tithof, R. O. Grigoriev, and M. F. Schatz, "Forecasting fluid flows using the geometry of turbulence," *Phys. Rev. Lett.*, vol. 118, p. 114 501, 11 2017.

[95] F. V. Dolzhanskii, V. A. Krymov, and D. Y. Manin, "Stability and vortex structures of quasi-two-dimensional shear flows," *Sov. Phys. Usp.*, vol. 33, no. 7, pp. 495–520, 1990.

[96] ——, "An advanced experimental investigation of quasi-two-dimensional shear flows," *J. Fluid Mech.*, vol. 241, pp. 705–722, Jul. 1992.

[97] N. F. Bondarenko, M. Z. Gak, and F. V. Dolzhanskiy, "Laboratory and theoretical models of plane periodic flows," *Izv. Akad. Nauk SSSR, Fiz. Atmos. Okeana*, vol. 15, no. 10, pp. 711–716, 1979.

[98] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[99] B. Suri, J. Tithof, R. O. Grigoriev, and M. F. Schatz, "Unstable equilibria and invariant manifolds in quasi-two-dimensional kolmogorov-like flow," *Phys. Rev. E*, vol. 98, p. 023 105, 2 2018.

[100] G. Kamath, "Bounds on the expectation of the maximum of samples from a gaussian," *URL http://www. gautamkamath. com/writings/gaussian max. pdf*, 2015.

[101] F. Orabona and D. Pal, *Optimal non-asymptotic lower bound on the minimax regret of learning with expert advice*, 2015. arXiv: `1511.02176 [stat.ML]`.

[102] G. Sivashinsky, "Nonlinear analysis of hydrodynamic instability in laminar flames: I. derivation of basic equations," *Acta astronautica*, vol. 4, pp. 1177–1206, 1977.

[103] Y. Kuramoto and T. Tsuzuki, "Persistent propagation of concentration waves in dissipative media far from thermal equilibrium," *Progress of theoretical physics*, vol. 55, no. 2, pp. 356–369, 1976.

[104] G. I. Sivashinsky and D. Michelson, "On irregular wavy flow of a liquid film down a vertical plane," *Progress of theoretical physics*, vol. 63, pp. 2112–2114, 1980.

[105] A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff pdes," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1214–1233, 2005.

[106] L. N. Trefethen and J. Weideman, "The exponentially convergent trapezoidal rule," *siam REVIEW*, vol. 56, no. 3, pp. 385–458, 2014.

[107] R. W. Wittenberg and P. Holmes, "Scale and space localization in the kuramoto–sivashinsky equation," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 9, no. 2, pp. 452–465, 1999.

[108] N. Kopell and L. N. Howard, "Plane wave solutions to reaction-diffusion equations," *Studies in Applied Mathematics*, vol. 52, no. 4, pp. 291–328, 1973.